RP04

FUNCTIONAL CONTROLLER

MD-11-DERPV-B

TEST (PART 2)

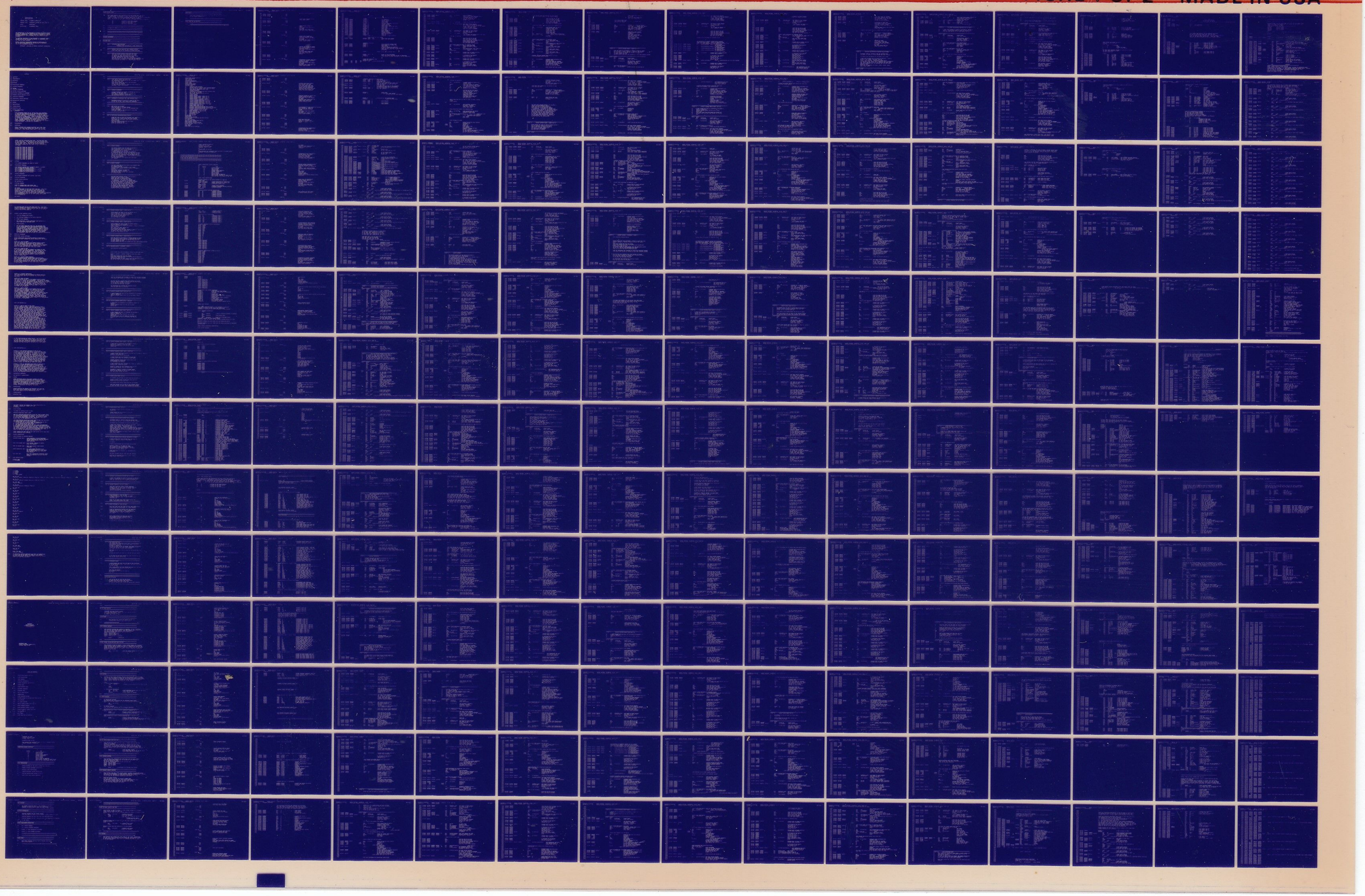
EP-DERPV-B-DL

FICHE 1 OF 2

MAY 1978

digital

MADE IN USA



RP04

FUNCTIONAL CONTROLLER

MD-11-DERPV-B

TEST (PART 2)

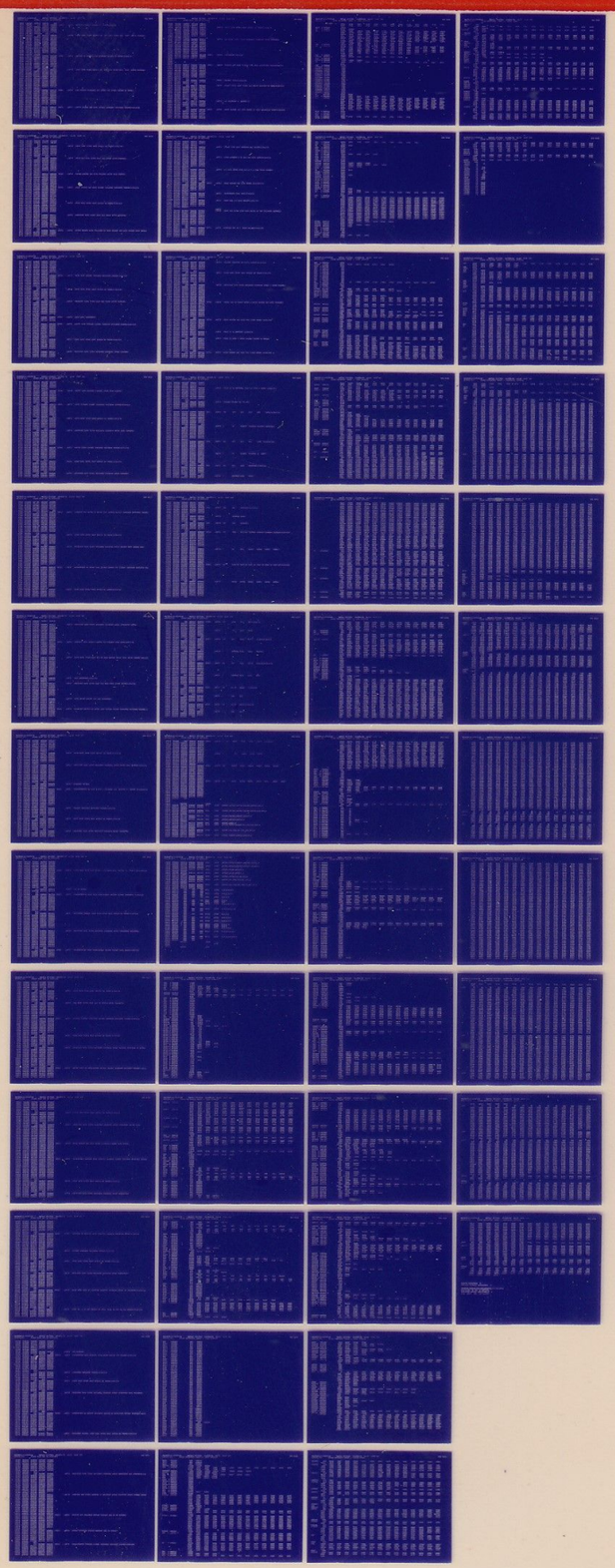
EP DERPV B DL

FICHE 2 OF 2

MAY 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DERPV-B-D

PRODUCT NAME: FUNCTIONAL CONTROLLER TEST (PART 2)

DATE: MARCH,1976

MAINTAINER: DIAGNOSTIC GROUP

CONTENTS

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE DCL OF THE RP04 DISK SUBSYSTEM. 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 SUBSYSTEM WORKS SUCCESSFULLY WHILE STANDING ALONE. SYSTEMS INTERACTION AND DRIVE TIMING IS LEFT TO OTHER DIAGNOSTICS. THIS IS WITH THE ASSUMPTION THAT STATIC 1 (DERPS AND DERPT) HAS BEEN RUN SUCCESSFULLY.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04 DISK SYSTEM. THE RP04 DISK SYSTEM WILL CONSIST OF AN RH70 CONTROLLER, A DISK CONTROL LOGIC (DCL), A DEC 733 DISK

DRIVE, AND ITS APPROPIATE 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 10, 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 10, SECTOR 21
CYLINDER 410,TRACK 10, SECTOR 21

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY

2.3 PRELIMINARY PROGRAMS

THIS PROGRAM ASSUMES THAT MAINDEC-11-DERPS- - (LATEST
REV) HAS BEEN RUN WITHOUT ERRORS.
AND IT ASSUMES THAT MAINDEC-11-DERPT- - (LATEST
REV) HAS BEEN RUN WITHOUT ERRORS.
AND IT ASSUMES THAT MAINDEC-11-DERPU- - (LATEST
REV) HAS BEEN RUN WITHOUT ERRORS.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

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

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS
STARTING ADDRESS ALL THE RP04S ON THE SYSTEM WILL BE
TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT.
TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE
THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RMAS REGISTER
THAT RESPONDS)THEN GO ON TO THE NEXT HIGHER UNIT NUMBER
THAT IS POWERED UP.

210 START

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

4.3 PROGRAM AND/OR OPERATOR ACTION

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

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

SWITCH 14 = LOOP ON TEST
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP ON THE CURRENT TEST BEING EXECUTED. FOR EXAMPLE IF THIS SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY. ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR DURING A PROGRAM HALT.

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

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

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 DEBUG, 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 128 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 SW06 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" RH06).

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES".

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE. THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04 REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

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.

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.

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.

DOCUMENT
****
MAINDEC-11-DERPVA-A
****

TABLE OF CONTENTS

```
                2       COPYRIGHT (C) 1974
                        DIGITAL EQUIPMENT CORP.
                        MAYNARD, MASS. 01754

                        PROGRAM BY SUB MALLICK

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


                ;*****************************************************************
       22       ;OPERATIONAL SWITCH SETTINGS
                ;*****************************************************************

                23
                                SWITCH                  USE
                                ------          --------------------
                                  15            HALT ON ERROR
                                  14            LOOP ON TEST
                                  13            INHIBIT ERROR TYPEOUTS
                                  11            INHIBIT ITERATIONS
                                  10            BELL ON ERROR
                                   9            LOOP ON ERROR
                                   8            LOOP ON TEST IN SWR<7:0>
                                   7            STOP FURTHER COMPARES IF SW08 IS LOW
                                   6            TYPE ALL REG. WITH ERROR IF SW8 LOW

                ;*****************************************************************
       36       ;BASIC DEFINITIONS
                ;*****************************************************************

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

                49      GENERAL PURPOSE REGISTER DEFINITIONS

                61      PRIORITY LEVEL DEFINITIONS

                71      "SWITCH REGISTER" SWITCH DEFINITIONS

                99      DATA BIT DEFINITIONS (BIT00 TO BIT15)

                127     BASIC "CPU" TRAP VECTOR ADDRESSES
```

```
                ;*********************************************************************
142             ;TRAP CATCHER
                ;*********************************************************************

145             ;       ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                ;       SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                ;       LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                ;*********************************************************************
149             ;STARTING ADDRESS(ES)
                ;*********************************************************************

157             ;       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

163             ;       AT THE DRIVE

                ;*********************************************************************
165             ;MEMORY MANAGEMENT DEFINITIONS
                ;*********************************************************************

167             ;       KT11 VECTOR ADDRESS

171             ;       KT11 STATUS REGISTER ADDRESSES

178             ;       KERNAL "I" PAGE DESCRIPTOR REGISTERS

189             ;       KERNAL "I" PAGE ADDRESS REGISTERS

200             ;       *****************************************************************

202             ;       *****************************************************************

                ;*********************************************************************
204             ;COMMON TAGS
                ;*********************************************************************

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

258             ;       *****************************************************************
```

```
                ;*********************************************************************
260             ;ERROR POINTER TABLE
                ;*********************************************************************

262             ;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:

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

277             ;*********************************************************************

964             ;*********************************************************************

970             ;*********************************************************************

1004            ;*********************************************************************
                ;*********************************************************************

1179            ;*********************************************************************

                ;*********************************************************************
1184            ;REGISTER ADDRESSES
                ;*********************************************************************

                ;*********************************************************************
1346            ;REGISTER TEST
                ;*********************************************************************

1466                    ;*************************************************************
                        ;TEST 1  REFERENCE EACH REGISTER
                                 REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                        ;*************************************************************

1504                    ;*************************************************************
                        ;TEST 2  PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
                                 CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES
                        ;*************************************************************

1524                    ;*************************************************************
                        ;TEST 3  TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                                 THE NUMBER OF RP04 DRIVES PRESENT ARE FOUND
                                 BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
                                 IN RHCS2 INCREMENTED FROM ZERO TO SEVEN
                                 THEN THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT
                                 THE DRIVE TYPE IS CHECKED TO HAVE 2020 OR 24020 AND THEN
                                 UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
```

```
1533    ;*****************************************************************

1644    ;*****************************************************************
        TEST 4  TYPE SERIAL NUMBER AND DRIVE TYPE
                SET APPROPIATE ATTTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
                TYPE UNIT UNDER TEST
                READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
                TYPE IT OUT AND PROCEED
                TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
                SWITCHES 0 THRU 7 AND RESTART

1653    ;*****************************************************************

1696    ;*****************************************************************
        TEST 5  PROGRAM INTERRUPT

1699            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
        ;*****************************************************************

1738    ;*****************************************************************
        TEST 6  INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

1741            PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
                IE AND RDY IS SET. THIS SHOULD NOT INTERRUPT
        ;*****************************************************************

1777    ;*****************************************************************

1778    TEST 7  SET VV BIT #6 IN RHDS1
                THIS TEST SETS VV IN RHDS1 INCASE
                ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
                IS NOT PERFORMED
                THERE IS A RESET AT THE BEGINING OF THIS TEST
                FOR ERROR RECOVERY ONLY.

1785    ;*****************************************************************

1902    ;*****************************************************************
        TEST 10 LAST BLOCK TRANSFERED-RHDS1 BIT#10

1905            WRITE ONE WORD OF 65125 ON CYLINDER 410, TRACK 18
                SECTOR 21, BY A WRITE HEADER AND DATA COMMAND
                THEN CHECK ALL REGISTERS (LAST BLOCK TRANSFERED
                SHOULD BE SET)
                THEN READ ABOVE BY READ DATA 256 WORDS
                AGAIN LBT SHOULD BE SET
                CHECK ALL REGISTERS AND DATA
```

```
1913    ********************************************************************

2172    ********************************************************************
        TEST 11 SEARCH COMMAND

2175            THE SEARCH COMMAND WILL BE DONE ON CYLINDER 0
                THAT IS STARTING WITH A RECALIBRATE
                THEN HEADER AND DATA WILL BE WRITTEN FOR SECTOR 0 AND 1
                ALL REGISTERS WILL BE CHECKED
                A SEARCH COMMAND WILL BE GIVEN FOR SECTOR 0
                ON INTERRUPT SECTOR 1 HEADER AND DATA WILL BE READ
                TIME WILL BE CRITICAL AS THE TIME TAKEN TO DO THE
                READ IS THE ONLY INDICATOR THAT THE HEADS WERE ON
                SECTOR 0 AT INTERRUPT TIME.  TIME ALLOWED IS MAXIMUM
                OF 1500 MICRO SECONDS
                THEN ALL REGISTERS ARE CHECKED AND DATA READ
                IS  CHECKED

2188    ********************************************************************

2535    ********************************************************************
        TEST 12 SEARCH COMMAND

2538            THE ONLY THING NEW IN THIS TEST IS AN IMPLIED SEEK
                IN A SEARCH COMMAND
                THE HEADS START FROM CYLINDER 10 BY A SEEK
                COMMAND THEN A SEARCH SECTOR 0 TRACK 0 CYLINDER 0
                IS GIVEN
                THEN A READ COMMAND IS GIVEN FOR
                CYLINDER 0, TRACK 0, SECTOR 1
                TIME FOR THE READ IS THE ONLY INDICATOR
                OF CORRECT SEARCH

2548    ********************************************************************

2772            THE NEXT TEST REMOVES SECTOR 1 ON CYLINDER 0
                TRACK0 AND PUTS SECTOR 0 THERE.
                HENCE THE PACK IS UNFORMATTED FROM
                THIS POINT ON TO THE TEST WHEN SECTOR
                1 IS REPLACED.  IF TESTING IS STOPPED WITH
                AN ERROR IN THE SECTION OF THE PROGRAM BETWEEN
                THIS AND WHEN SECTOR 1 IS REPLACED THEN THE
                DISK BEING USED MAY HAVE BEEN UNFORMATTED
                IF THE LAST PASS OF THIS PROGRAM GIVES
                NO ERRORS IN THIS SECTION THEN THE DISK
                MAY NOT HAVE BEEN UNFORMATTED. HOWEVER IT
                IS RECOMMENDED THAT AFTER A PASS OF THIS
                PROGRAM THE DISK BE REFORMATTED.
```

```
2790    ********************************************************************
        TEST 13 HEADER COMPARE ERROR - RHER1 BIT #7

2793            WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
                AND PUT SECTOR 0 THERE ON CYLINDER 0
                THEN A READ DATA IS GIVEN FOR SECTOR1
                HCE- BIT #7 IN RHER1 SHOULD SET.
                ALL REGISTERS ARE CHECKED
                ANY DATA READ IS CHECKED

2800    ********************************************************************

3030    ********************************************************************
        TEST 14 HEADER COMPARE ERROR - RHER1 BIT #7

3033            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 70. 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

3043    ********************************************************************

3450    ********************************************************************
        TEST 15 HEADER COMPARE ERROR - RHER1 BIT #7

3453            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

3459    ********************************************************************

3606    ********************************************************************
        TEST 16 RESTORE SECTOR 1 CYLINDER 1 TRACK 1

3609            THIS REPLACES REMOVER SECTOR


                WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD

3614            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.

3635    ;***************************************************************

3943    ;***************************************************************
        TEST 17 INVALID ADDRESS ERROR RHER1-BIT #10

3946                    A WRITE HEADER AND DATA WILL BE ATTEMPTED TO CYLINDER 411
                        TRACK 0, SECTOR 0
                        INVALID ADDRESS ERROR (IAE) BIT #10 IN RHER1 SHOULD
                        SET

3951    ;***************************************************************

4090    ;***************************************************************
        TEST 20 INVALID ADDRESS ERROR RHER1 - BIT #10

4093                    A WRITE DATA IS ATTEMPTED TO CYLINDER 0, TRACK 19,
                        SECTOR 0
                        INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
                        SHOULD SET

4098    ;***************************************************************

4226    ;***************************************************************
        TEST 21 INVALID ADDRESS ERROR RHER1 -BIT #10

4229                    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

4237    ;***************************************************************
```

```
4399    ********************************************************************
        TEST 22 INVALID ADDRESS ERROR RHER1 - BIT #10

4402            A READ DATA IS ATTEMPTED TO CYLINDER 0, TRACK 0
                SECTOR 20 - FORMAT 18 BITS PER WORD
                INVALID ADDRESS ERROR IAF BIT #10 IN RHER1
                SHOULD SET

4407    ********************************************************************

4532    ********************************************************************
        TEST 23 ADDRESS OVERFLOW ERROR RHER1- BIT#9 AOE

4535            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, TRACK 18, SECTOR 21, 261 WORDS

                ADDRESS OVERFLOW ERROR RHER1 BIT#9 SHOULD SET
                AFTER SECTOR 21 IS WRITTEN
                ALL REGISTERS ARE CHECKED

                A READ HEADER AND DATA CYLINDER 410, TRACK 18, SECTOR 21,
                260+66+4=330  WORDS ARE GIVEN

                SECTOR 21 SHOULD BE READ CORRECTLY BUT NO MORE
                READS SHOULD HAPPEN, AOE BIT SHOULD SET

                CYLINDER 0, TRACK 0, SECTOR 0 IS READ THERE
                SHOULD BE NO CHANGE IN DATA IN THIS SECTOR BY
                THE LAST WRITE HEADER AND DATA

4556    ********************************************************************

5122    ********************************************************************
        TEST 24 FORMAT ERROR RHER1 BIT #4

5125            AN ATTEMPT WILL BE MADE TO WRITE DATA ON CYLINDER 0

5126            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
        ********************************************************************
```

```
5470    ********************************************************************
        TEST 25 FORMAT ERROR RHER1 BIT #4

5473            AN ATTEMPT IS MADE TO READ DATA WITH WRONG
                FORMAT BIT

                FORMAT ERROR BIT #4 IN RHER1 SHOULD SET
                NO DATA SHOULD BE READ

5479    ********************************************************************

5636    ********************************************************************
        TEST 26 REGISTER MODIFICATION REFUSED BIT #2 RHER1

5639            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

5654    ********************************************************************

5931    ********************************************************************
        TEST 27 REGISTER MODIFICATION REFUSED RMR-BIT#2 IN RHER1

5934            A WRITE HEADER AND DATA COMMAND WILL BE GIVEN TO
                CYLINDER 1 SECTOR 0 TRACK 0 DATA WORDS

5936            OF 070707

                A WRITE DATA COMMAND WILL BE GIVEN TO CYLINDER 1

5940            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
```

```
5953    ;*****************************************************************
6338    ;*****************************************************************
        TEST 30 REGISTER MODIFICATION REFUSED BIT #2 RHER1

6341            A READ DATA COMMAND IS GIVEN TO CYLINDER 0, SECTOR 0
                TRACK 0, IMMEDIATELY AFTER GO RHAS IS WRITTEN INTO
                WITH ALL ONES RMR BIT #2 IN RHER SHOULS NOT SET

6345    ;*****************************************************************
6493    ;*****************************************************************
        TEST 31 ILLEGAL FUNCTION BIT #0 IN RHER1

6496            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

6502    ;*****************************************************************
6676    ;*****************************************************************
        TEST 32 ERROR REGISTER 1 - BIT #13 OPI
                A WRITE HEADER AND DATA COMMAND IS GIVEN
                CYLINDER 0 SECTOR 1 TRACK 0 KEYS 0 DATA 177777
                WORDCOUNT 260

                AFTER GO IS GIVEN THEN THREE INDEX PULSES ARE
                GIVEN, THIS SHOULD BRING OPI HIGH
        ;*****************************************************************
6842    ;*****************************************************************
        TEST 33 ERROR REGISTER #1, BIT #13 OPI

6845            THIS WILL TEST THAT OPI DOES NOT SET WHEN THREE NORMAL
                INDEX PULSES ARE ENCOUNTERED IN A READ COMMAND

6848            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


6854            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.
```

```
6861    ********************************************************************

7043    ********************************************************************
        TEST 34 HEAD SELECTION TEST
                THIS TESTS HEAD SELECTION LOGIC ONLY, A WRITE HEADER AND
                DATA COMMAND IS GIVEN TO EACH TRACK FROM 0 TO 10 ON
                CYLINDER 0, SECTOR 0.
                THE DATA ON EACH SECTOR IS UNIQUE, THE LEAST SIGNIFICANT
                5 BITS GIVE SECTOR THE NEXT LEAST SIGNIFICANT 5 BITS
                GIVE TRACK THE NEXT 6 BITS GIVE CYLINDER

                THEN READ HEADER AND DATA IS DONE FOR THE ABOVE AND DATA
                CHECKED

                BETWEEN THE WRITE AND READ ONLY ERR AND TRE ARE CHECKED


                ON AN ERROR IN THE READ HEADER AND DATA LOOPING WILL BE
                ONLY ON THE ERROR SECTOR READ

7061    ********************************************************************

7345    ********************************************************************
        TEST 35 DIFFERENCE LINES

7348            A WRITE HEADER AND DATA WILL BE DONE ON ALL CYLINDERS
                UP TO 256 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.

                DATA WILL BE CHECKED
                ON AN ERROR LOOPING WILL BE DONE ON
                READ ONLY

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

7652    ********************************************************************
        ********************************************************************
        TEST 36 END OF DRIVE

7656            THIS IS THE END OF TEST FOR ONE DRIVE
                IF THERE ARE MORE DRIVES THEN THE PROGRAM
                JUMPS TO TEST 5 FOR NEXT DRIVE TEST
                END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE

7661    ********************************************************************
```

```
                7697    ;*********************************************************************

                ;*********************************************************************
        7699    END OF PASS ROUTINE
                ;*********************************************************************

                7701    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

                ;*********************************************************************
        7741    SUBROUTINES
                ;*********************************************************************

                8407            THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                                ADDRESS FROM 176700 TO ANY TYPED VALUE

                8474    ;*****************************
                        ;*********************************************************************

                ;*********************************************************************
        8477    SCOPE HANDLER ROUTINE
                ;*********************************************************************

                8479    THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                        AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        SW14=1  LOOP ON TEST
                        SW11=1  INHIBIT ITERATIONS
                        SW09=1  LOOP ON EPROR
                        SW08=1  LOOP ON TEST IN SWR<7:0>
                        CALL
                                SCOPE           ;;SCOPE=IOT

                8538    ;*********************************************************************

                ;*********************************************************************
        8540    CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                ;*********************************************************************

                8542    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
```

```
8606    ;*********************************************************************

        ;*********************************************************************
8608    ;TYPE ROUTINE
        ;*********************************************************************

8610    ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
        ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
        ;NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
        ;NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
        ;NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

        ;CALL:
        ;1) USING A TRAP INSTRUCTION
        ;       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
        ;OR
        ;       TYPE
        ;       MESADR

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

8653    ;*********************************************************************

        ;*********************************************************************
8655    ;TTY INPUT ROUTINE
        ;*********************************************************************

8664    ;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

8681    ;TK SERVICE ROUTINE
        ;THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT

8703    ;*********************************************************************
        ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
        ;CALL:
        ;       RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
        ;       RETURN HERE             ;;CHARACTER IS ON THE STACK

8723    ;*********************************************************************
        ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
        ;CALL:
        ;       RDLIN                   ;;INPUT A STRING FROM THE TTY
        ;       RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
        ;                               ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
8759    ;*********************************************************************

        ;*********************************************************************
8761    ;READ AN OCTAL NUMBER FROM THE TTY
        ;*********************************************************************

8763    ;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

8813    ;*********************************************************************

        ;*********************************************************************
8815    ;ERROR HANDLER ROUTINE
        ;*********************************************************************

8817    ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
        ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
        ;AND GO TO $ERRTYP ON ERROR
        ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
        ;SW15=1  HALT ON ERROR
        ;SW13=1  INHIBIT ERROR TYPEOUTS
        ;SW10=1  BELL ON ERROR
        ;SW09=1  LOOP ON ERROR
        ;CALL
                ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

        ;*********************************************************************
8862    ;ERROR MESSAGE TYPEOUT ROUTINE
        ;*********************************************************************

8863    ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
        ;ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
        ;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
        ;IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.

8867    ;WITH ONLY MINOR CHANGES
        ;FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
        ;ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
        ;SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
        ;AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
        ;AND NOT THE ERROR MESSAGE AND HEADER.
```

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

                ;*********************************************************************
;9092            ;BINARY TO OCTAL (ASCII) AND TYPE
                ;*********************************************************************

                ;9094    THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                ;        OCTAL (ASCII) NUMBER AND TYPE IT.
                ;        $TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                ;        CALL:
                ;                MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;                TYPOS                   ;;CALL FOR TYPEOUT
                ;                .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                ;                .BYTE   M               ;;M=1 OR 0
                ;                                        ;;1=TYPE LEADING ZEROS
                ;                                        ;;0=SUPPRESS LEADING ZEROS

                ;        $TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                ;        $TYPOS OR $TYPOC
                ;        CALL:
                ;                MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;                TYPON                   ;;CALL FOR TYPEOUT

                ;        $TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                ;        CALL:
                ;                MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;                TYPOC                   ;;CALL FOR TYPEOUT

                ;9168    *********************************************************************

                ;*********************************************************************
;9170            ;TRAP DECODER
                ;*********************************************************************

                ;9172    THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                ;        AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                ;        OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                ;        GO TO THAT ROUTINE.
```

```
                ;*********************************************************************
9185            ;TRAP TABLE
                ;*********************************************************************

                ;9187    THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                ;        BY THE "TRAP" INSTRUCTION.

                ;9205    *********************************************************************

                ;*********************************************************************
9207            ;POWER DOWN AND UP ROUTINES
                ;*********************************************************************

                ;9247    *************************************************************************
                ;        *************************************************************************
```

```
                        .TITLE  MAINDEC-11-DERPV-B
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY SUB MALLICK
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-A5).
                        ;*

;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;
;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;
;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;/*\;
;DRIVE MUST BE LOCKED IN A OR B
;/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*
;/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*
;/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*/;\*


                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------                  --------------------
                        ;*        15                    HALT ON ERROR
                        ;*        14                    LOOP ON TEST
                        ;*        13                    INHIBIT ERROR TYPEOUTS
                        ;*        11                    INHIBIT ITERATIONS
                        ;*        10                    BELL ON ERROR
                        ;*         9                    LOOP ON ERROR
                        ;*         8                    LOOP ON TEST IN SWR<7:0>
                        ;*         7                    STOP FURTHER COMPARES IF SW08 IS LOW
                        ;*         6                    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
        177776          PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
        177774          STKLMT= 177774          ;;STACK LIMIT REGISTER
        177772          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
        177570          SWR=    177570          ;;SWITCH REGISTER
        177570          DISPLAY=SWR

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

```
                        R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIOPITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
```

```
                    000004          BIT02=  4
                    000002          BIT01=  2
                    000001          BIT00=  1
                                    .EQUIV  BIT09,BIT9
                                    .EQUIV  BIT08,BIT8
                                    .EQUIV  BIT07,BIT7
                                    .EQUIV  BIT06,BIT6
                                    .EQUIV  BIT05,BIT5
                                    .EQUIV  BIT04,BIT4
                                    .EQUIV  BIT03,BIT3
                                    .EQUIV  BIT02,BIT2
                                    .EQUIV  BIT01,BIT1
                                    .EQUIV  BIT00,BIT0

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


                                    .SBTTL  TRAP CATCHER

                    000000                  .=0
                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                                    .SBTTL  STARTING ADDRESS(ES)
                    000200                  .=200

000200  000137  004666                      JMP     @#BEGIN         ;;JUMP TO STARTING ADDRESS OF PROGRAM
                    000210                  .=210
000210  000137  004652                      JMP     @#BEGIN2        ;JUMP SELECT TEST
                    000220                  .=220
000220  000137  004636                      JMP     @#BEGIN1        ;JUMP TO NO OPERATOR TESTS
                                    ;*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
```

```
                        MMVEC=  250

                        ;*KT11 STATUS REGISTER ADDRESSES

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

                        ;*KERNAL "I" PAGE DESCRIPTOR REGISTERS

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

                        ;*KERNAL "I" PAGE ADDRESS REGISTERS

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

                        ;*****************************************************************
                                .=1110
```

```
                                ;*****************************************************************
                                .SBTTL  COMMON TAGS

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

                    000046              .=46
000046  027650                $ENDAD                  ;;LOGICAL END OF PROGRAM

                    001100              .=1100

001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000                 $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
001102     000                  $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103     000                  $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000                 $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                 $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP
001110  000000                 $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                 $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114     000                  $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115     001                  $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                 $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                 $GDADR: .WORD   0       ;;CONTAINS  OF 'GOOD' DATA
001122  000000                 $BDADR: .WORD   0       ;;CONTAINS  OF 'BAD' DATA
001124  000000                 $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000                 $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000  000000  000000         .WORD   0,0,0   ;;RESERVED--NOT TO BE USED
001136  177560                 $TKS:   177560          ;;TTY KBD STATUS
001140  177562                 $TKB:   177562          ;;TTY KBD BUFFER
001142  177564                 $TPS:   177564          ;;TTY PRINTER STATUS REG.
001144  177566                 $TPB:   177566          ;;TTY PRINTER BUFFER REG.
001146     000                  $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001147     002                  $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150     012                  $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001151     000                  $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152  000000                 $REGAD: .WORD   0       ;;CONTAINS THE  FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001154  000000                 $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
001156  000000                 $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
001160  000000                 $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
001162  000000                 $REG3:  .WORD   0       ;;CONTAINS (($REGAD)+6)
001164  000000                 $REG4:  .WORD   0       ;;CONTAINS (($REGAD)+10)
001166  000000                 $REG5:  .WORD   0       ;;CONTAINS (($REGAD)+12)
001170  000000                 $TMP0:  .WORD   0       ;;USER DEFINED
001172  000000                 $TMP1:  .WORD   0       ;;USER DEFINED
001174  000000                 $TMP2:  .WORD   0       ;;USER DEFINED
001176  000000                 $TMP3:  .WORD   0       ;;USER DEFINED
001200  000000                 $TMP4:  .WORD   0       ;;USER DEFINED
001202  000000                 $TMP5:  .WORD   0       ;;USER DEFINED
001204  000000                 $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
001206  000000                 $ESCAPE:0               ;;ESCAPE ON ERROR
001210  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001214     077                  $QUES:  .ASCII  /?/     ;;QUESTION MARK
001215     015                  $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001216  000012                 $LF:    .ASCIZ  <12>    ;;LINE FEED
```

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

                        .SBTTL  ERROR POINTER TABLE

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

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

001220                  $ERRTB:


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

                        ;ITEM1
001220  036452                  EM1                     ;RP04 DID NOT INTERRUPT
                                                        ;WAITED ON BIT DID NOT OCCUR
001222  052666                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001224  054706                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001226  055170                  DF1                     ;0,0,0,0,0,0

                        ;ITEM2
001230  036501                  EM2                     ;INTERRUPT ENABLE BIT DOWN BUT
                                                        ;WAITED ON BIT DID NOT OCCUR
001232  052666                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001234  054706                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001236  055170                  DF1                     ;0,0,0,0,0,0

                        ;ITEM3
001240  036570                  EM3                     ;RP04 DID NOT INTERRUPT WHEN
                                                        ;WAITED ON BIT DID SET
001242  052666                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;RHCS1 CONTENTS
001244  054706                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
```

```
                                                DF1             ;0,0,0,0,0,0
                                        ;ITEM4
                                                EM4             ;WAITED ON BIT DID SET BUT
                                                                ;TIME IS IN ERROR
                                                                ;TIME IS GIVEN IN 10 MICRO SEC.
                                                                ;(DECIMAL)
                                                DH4             ;PC
                                                                ;WAT PC
                                                                ;BIT WAITED
                                                                ;REG ADDRESS
                                                                ;TIME IN 10 MSEC
                                                DT4             ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,WAITIM
                                                DF4             ;0,0,0,0,0,1
                                        ;ITEM5
                                                EM5             ;RHAS DOES NOT CLEAR BY
                                                                ;MOVING IN ALL ONES
                                                DH5             ;PC
                                                                ;REG. ADDR.
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA
                                                DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
                                                DF5             ;0,0,0,0
                                        ;ITEM6
                                                EM6             ;LOADING RHER1 FOR ALL
                                                                ;UNITS DID NOT SET ANY BITS
                                                                ;IN RHAS-NO UNITS PRESENT
                                                DH6             ;PC
                                                                ;REG ADDR
                                                                ;RECEIVED DATA
                                                DT6             ;$ERRPC,REGADR,$BDDAT
                                                DF6             ;0,0,0
                                        ;ITEM7
                                                EM7             ;SPECIFIED REGISTER NONEXISTANT
                                                                ;SO ABORT PROGRAM
                                                DH7             ;PC
                                                                ;ADDR. OF REG.
                                                DT7             ;$ERRPC,TEMP1
                                                DF7             ;0,0
                                        ;ITEM10
                                                EM10            ;STOPED DRIVE HAS MOL BIT
                                                                ;IN RHDS1 = 1
                                                DH10            ;PC
                                                                ;TEST NO
                                                                ;FAILING REG ADDR
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHCS2
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1
                                                DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
                                                DF10            ;0,0,0,0,0,0,0
```

```
                                ;ITEM11
001320  037251                          EM11            ;WITH SPINDLE POWERED DOWN
                                                        ;RHCS2 SHOULD HAVE ONLY
                                                        ;UNIT NUMBER AND IR HIGH
001322  053357                          DH10            ;PC
                                                        ;TEST NO
                                                        ;FAILING REG. ADR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001324  054772                          DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001326  055215                          DF10            ;0,0,0,0,0,0,0

                                ;ITEM12
001330  037356                          EM12            ;AFTER A POWER UP WITH
                                                        ;NO PACK ACKNOWLEDGE COMMAND
                                                        ;RHDS1 SHOULD HAVE MOL=1, VV=0
001332  053357                          DH10            ;PC
                                                        ;TEST NO
                                                        ;FAILING REGISTER ADDR.
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTETNS OF RHDS1
                                                        ;CONTENTS OF RHER1
001334  054772                          DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001336  055215                          DF10            ;0,0,0,0,0,0,0

                                ;ITEM13
001340  037464                          EM13            ;AFTER A POWER UP WITHOUT
                                                        ;ANY INIT RHCS1 SHOULD
                                                        ;HAVE GO=0, DVA=1, RDY=1
                                                        ;IE=0, DISREGARD
                                                        ;ALL OTHER BITS
001342  053357                          DH10            ;PC
                                                        ;TEST NO
                                                        ;FAILING REGISTER ADDR.
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001344  054772                          DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001346  055215                          DF10            ;0,0,0,0,0,0,0

                                ;ITEM14
001350  037603                          EM14            ;AFTER POWER UP RHCC
                                                        ;SHOULD BE=0
001352  053151                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001354  054742                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001356  055204                          DF5             ;0,0,0,0

                                ;ITEM15
001360  037655                          EM15            ;PACK ACKNOWLEDGE CAUSED
```

```
                                                        ;AN ERROR
                                                        ;GOOD DATA IS BEFORE COMMAND
                                                        ;RECEIVED DATA IS AFTER COMMAND
001362  053151                      DH5                   ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001364  054742                      DT5                   ;$ERRPC,REGADR,$GDDAT,$BDDAT
001366  055204                      DF5                   ;0,0,0,0

                            ;ITEM16
001370  040016                      EM16                  ;GIVING A NO-OP COMMAND CAUSED
                                                        ;AN ERROR
                                                        ;GOOD DATA GIVES REGISTER
                                                        ;CONTENTS BEFORE COMMAND
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER COMMAND
001372  053151                      DH5                   ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001374  054742                      DT5                   ;$ERRPC,REGADR,$GDDAT,$BDDAT
001376  055204                      DF5                   ;0,0,0,0

                            ;ITEM17
001400  040144                      EM17                  ;DRIVE CLEAR COMMAND
                                                        ;CAUSED AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
001402  053151                      DH5                   ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001404  054742                      DT5                   ;$ERRPC,REGADR,$GDDAT,$BDDAT
001406  055204                      DF5                   ;0,0,0,0

                            ;ITEM20
001410  040301                      EM20                  ;READ-IN COMMAND GAVE AN ERROR
                                                        ;GOOD DATA HAS WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA HAS WHAT WAS
                                                        ;AFTER COMMAND
001412  053151                      DH5                   ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001414  054742                      DT5                   ;$ERRPC,REGADR,$GDDAT,$BDDAT
001416  055204                      DF5                   ;0,0,0,0


                            ;ITEM 21
001420  040450                      EM21                  ;RHCS1 CONTENTS DURING
                                                        ;COMMAND WAS IN ERROR
001422  053151                      DH5
001424  054742                      DT5
```

```
                                        DF5
                                ;ITEM 22
                                        EM22            ;RHDS1 CONTENTS DURING
                                                        ;COMM ANS WAS IN ERROR
                                        DH5
                                        DT5
                                        DF5
                                ;ITEM 23
                                        EM23            ;UNLOAD COMMAND GAVE AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
                                        DH5
                                        DT5
                                        DF5
                                ;ITEM 24
                                        EM24            ;OFFSET COMMAND CAUSED AN ERROR
                                                        ;GOOD DATA IS WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
                                                        ;AFTER AN OFFSET COMMAND
                                        DH5
                                        DT5
                                        DF5
                                ;ITEM 25
                                        EM25            ;RETURN TO CENTER LINE COMMAND
                                                        ;CAUSED AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
                                        DH5
                                        DT5
                                        DF5
                                ;ITEM 26
                                        EM26            ;500 OFFSETS CAUSED AN ERROR
                                        DH26            ;PC
                                                        ;CONT. OF RHCS1
                                                        ;CONT. OF RHCS2
                                                        ;CONT. OF RHDS1
                                                        ;CONT. OF RHER1
                                                        ;CONT. OF RHER2
                                                        ;CONT. OF RHER3
                                        DT26            ;$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
                                        DF26            ;0,0,0,0,0,0,0
                                ;ITEM 27
                                        EM27            ;WRITE HEADER AND DATA
                                                        ;CAUSED IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT
                                                        ;SHOULD BE THERE
```

```
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER COMMAND
001502  053151                      DH5
001504  054742                      DT5
001506  055204                      DF5

                            ;ITEM 30
001510  041600                      EM30                ;WRITE HEADER AND DATA
                                                        ;CHANGED WRITE FROM BUFFER
001512  053715                      DH30                ;PC
                                                        ;WORD NO
                                                        ;GOOD DATA
                                                        ;BAD DATA
001514  055032                      DT30                ;$ERRPC,ERWORD,$GDDAT,$BDDAT
001516  055233                      DF30                ;0,0,0,0

                            ;ITEM 31
001520  041660                      EM31                ;READ HEADER AND DATA CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA HAS WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER COMMAND
001522  053151                      DH5
001524  054742                      DT5
001526  055204                      DF5

                            ;ITEM 32
001530  042075                      EM32                ;WRITE HEADER AND DATA FOLLOWED
                                                        ;BY A READ HEADER AND DATA
                                                        ;CAUSED A READ/WRITE ERROR
001532  053715                      DH30
001534  055032                      DT30
001536  055233                      DF30

                            ;ITEM 33
001540  042202                      EM33                ;READ DATA CAUSED IMPROPER REGISTER
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
                                                        ;COMMAND
001542  053151                      DH5
001544  054742                      DT5
001546  055204                      DF5

                            ;ITEM 34
001550  042404                      EM34                ;READ DATA INCORRECT
001552  053715                      DH30
001554  055032                      DT30
001556  055233                      DF30

                            ;ITEM 35
001560  042430                      EM35                ;WRITE DATA COMMAND CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES REGISTER
```

```
                                                        ;CONTENTS AFTER WRITE DATA
001562  053151                      DH5
001564  054742                      DT5
001566  055204                      DF5

                                ;ITEM 36
001570  042646                      EM36                ;WRITE DATA COMMAND CHANGED
                                                        ;WRITE FROM BUFFER
001572  053715                      DH30
001574  055032                      DT30
001576  055233                      DF30

                                ;ITEM 37
001600  042723                      EM37                ;SEEK COMMAND CAUSED AN
                                                        ;ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER SEEK COMMAND
001602  053151                      DH5                 ;
001604  054742                      DT5                 ;
001606  055204                      DF5                 ;

                                ;ITEM 40
001610  043140                      EM40                ;WRITE CHECK CAUSED AN
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
001612  053151                      DH5
001614  054742                      DT5
001616  055204                      DF5

                                ;ITEM 41
001620  043347                      EM41                ;LOCKING OUT WRITES BY WRITE
                                                        ;LOCK BUTTON CAUSED IMPROPER
                                                        ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER WRITES
                                                        ;WERE LOCKED OUT BY
                                                        ;BUTTON
001622  053151                      DH5
001624  054742                      DT5
001626  055204                      DF5

                                ;ITEM 42
001630  043630                      EM42                ;ATTEMPTING TO WRITE WITH WRITE
                                                        ;LOCKED OUT CAUSED IMPROPER
                                                        ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER ATTEMPT
```

```
001632  053151                          DH5
001634  054742                          DT5
001636  055204                          DF5

                                ;ITEM 43
001640  044106                          EM43            ;WRITING WITH WRITE LOCKED
                                                        ;OUT CHANGED DISK DATA
                                                        ;GOOD DATA GIVES WHAT WAS
                                                        ;ON DISK BEFORE WRITE WITH
                                                        ;WRITE LOCK WAS ATTEMPTED
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;READ BACK AFTER WRITE WITH
                                                        ;WRITE LOCK WAS ATTEMPTED
001642  053715                          DH30
001644  055032                          DT30
001646  055233                          DF30

                                ;ITEM 44
001650  044444                          EM44            ;ENABLING WRITES BY WRITE LOCK
                                                        ;BUTTON CAUSED AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER WRITE LOCK
                                                        ;BUTTON ENABLED WRITES
001652  053151                          DH5             ;
001654  054742                          DT5             ;
001656  055204                          DF5             ;

                                ;ITEM 45
001660  044736                          EM45            ;TRANSFERRING ON LAST BLOCK IE. CYLINDER
                                                        ;410, SECTOR 21, TRACK 10
                                                        ;CAUSED IMPROPER REGISTER
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER TRANSFER
001662  053151                          DH5
001664  054742                          DT5
001666  055204                          DF5

                                ;ITEM 46
001670  045232                          EM46            ;DATA READ FROM LAST
                                                        ;BLOCK IE. CYLINDER 410
                                                        ;SECTOR 21, TRACK 10 IS IN
                                                        ;ERROR
001672  053715                          DH30
001674  055032                          DT30
001676  055233                          DF30

                                ;ITEM 47
001700  045345                          EM47            ;TRANSFERRING FROM NONEXISTANT
                                                        ;SECTOR CAUSED IMPROPER
                                                        ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
```

```
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER ATTEMPTED
                                                        ;TRANSFER
001702  053151                    DH5
001704  054742                    DT5
001706  055204                    DF5

                                ;ITEM 50
001710  045627                    EM50                 ;TRANSFERRING FROM NONEXISTANT
                                                        ;SECTOR CAUSED DATA ERROR
                                                        ;GOOD DATA GIVES WHAT
                                                        ;SHOULD BE IN BUFFER
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;IN BUFFER AFTER TRANSFER
001712  053715                    DH30
001714  055032                    DT30
001716  055233                    DF30

                                ;ITEM 51
001720  046046                    EM51                 ;GIVING ILLEGAL FUNCTION CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER ILLEGAL FUNCTION
001722  054007                    DH51                 ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
                                                        ;ILLEGAL FUNCTION
001724  055044                    DT51                 ;$ERRPC,REGADR,$GDDAT,$BDDAT,ILLEGL
001726  055237                    DF51                 ;0,0,0,0,0

                                ;ITEM 52
001730  046313                    EM52                 ;WRITE DATA ON NONEXISTANT
                                                        ;SECTOR CAUSED IMPROPER
                                                        ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER ATTEMPTED
                                                        ;WRITE DATA
001732  053151                    DH5
001734  054742                    DT5
001736  055204                    DF5

                                ;ITEM 53
001740  046564                    EM53                 ;READ HEADER AND DATA AFTER
                                                        ;A SEARCH CAUSED AN ERROR
001742  053715                    DH30
001744  055032                    DT30
001746  055233                    DF30

                                ;ITEM 54
```

```
                                        EM54            ;ATTEMPTED OPERATION WITH
                                                        ;INVALID ADDRESS CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER OPERATION
001752  053151                    DH5
001754  054742                    DT5
001756  055204                    DF5

                                ;ITEM 55
001760  047117                    EM55                ;WRITING/READING WITH EXPECTED
                                                        ;ADDRESS OVERFLOW ERROR CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER OPERATION
001762  053151                    DH5
001764  054742                    DT5
001766  055204                    DF5

                                ;ITEM 56
001770  047405                    EM56                ;DATA READ WITH AN EXPECTED
                                                        ;ADDRESS OVERFLOW ERROR IS
                                                        ;INCORRECT
                                                        ;WORD NO 1 TO 260 SHOULD
                                                        ;BE READ
                                                        ;WORD NOS 261 TO 266 SHOULD
                                                        ;NOT CHANGE DUE TO READ
001772  053715                    DH30
001774  055032                    DT30
001776  055233                    DF30

                                ;ITEM 57
002000  047615                    EM57                ;ATTEMPTING DATA COMMAND
                                                        ;WITH WRONG FORMAT BIT CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER ATTEMPTED DATA
                                                        ;TRANSFER
002002  053151                    DH5
002004  054742                    DT5
002006  055204                    DF5

                                ;ITEM 60
002010  050107                    EM60                ;ATTEMPTING TO MODIFY REGISTER
                                                        ;DURING AN OPERATION CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER OPERATION
```

```
                                                        ;WAS COMPLETE
002012  054126                          DH60            ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
                                                        ;MODFING REGISTER
002014  055060                          DT60            ;$ERRPC,REGADR,$GDDAT,$BDDAT,$BDADR
002016  055244                          DF60            ;0,0,0,0,0

                                ;ITEM 61
002020  050516                          EM61            ;DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
002022  054243                          DH61            ;PC
                                                        ;PC OF JSR
                                                        ;RHDS1
002024  055074                          DT61            ;$ERRPC,PCJSR,$BDADR
002026  055251                          DF61            ;0,0,0

                                ;ITEM 62
002030  050516                          EM61            ;DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
002032  054326                          DH62            ;PC
                                                        ;PC OF JSR
                                                        ;RHCS1 WAS
002034  055104                          DT62            ;$ERRPC,PCJSR,$BDADR
002036  055254                          DF62            ;0,0,0


                                ;ITEM 63
002040  050602                          EM63            ;RHDS1 CONTENTS DURING
                                                        ;COMMAND WAS IN ERROR
002042  053151                          DH5
002044  054742                          DT5
002046  055204                          DF5

                                ;ITEM 64
002050  050655                          EM64            ;RECALIBRATE COMMAND CAUSED
                                                        ;IMPROPER REGISTER CHANGE.
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE.
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
                                                        ;AFTER COMMAND
002052  053151                          DH5
002054  054742                          DT5
002056  055204                          DF5

                                ;ITEM65

002060  051074                          EM65            ;INTERRUPT FAILING
002062  054401                          DH65            ;PC
                                                        ;TEST NO
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHAS
                                                        ;CONTENTS OF RHDS1
002064  055114                          DT65            ;$ERRPC,TSTNM,CS1,AS,DS1
002066  055257                          DF65            ;0,0,0,0,0
```

```
                                ;ITEM66
002070  051116                          EM66            ;HEADER AND DATA COMMAND
                                                        ;FOR HEAD SELECTION TEST
                                                        ;CAUSED AN ERROR
                                                        ;RHDST GIVES WHAT TRACK
                                                        ;WAS BEING WRITTEN ON CYLINDER 0
                                                        ;SECTOR 0
002072  054522                          DH66            ;PC
                                                        ;RHDST
                                                        ;RHER1
                                                        ;RHER2
                                                        ;RHER3
                                                        ;RHCS1
                                                        ;RHCS2
002074  055130                          DT66            ;$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
002076  055264                          DF66            ;0,0,0,0,0,0,0
                                ;ITEM67
002100  051327                          EM67            ;READ HEADER AND DATA ERROR
                                                        ;IN HEAD SELECTION TEST
                                                        ;FIRST FOUR WORDS GIVE HEADER
                                                        ;NEXT WORDS ARE DATA
                                                        ;GOOD DATA WORDS GIVE
                                                        ;THE TRACK NUMBER IN
                                                        ;BITS 4,5,6,7,8
002102  053715                          DH30
002104  055032                          DT30
002106  055233                          DF30
                                ;ITEM70
002110  051603                          EM70            ;READ HEADER AND DATA ERROR
                                                        ;IN DIFFERENCE LINE TEST
                                                        ;WORD NOS. 1-4 GIVE
                                                        ;HEADER
                                                        ;WORD NOS. 5-260 GIVE DATA
                                                        ;WHICH IS THE CYLINDER
                                                        ;ADDRESS
002112  053715                          DH30
002114  055032                          DT30
002116  055233                          DF30
                                ;ITEM 71
002120  052011                          EM71            ;FORCING OPI CAUSED IMPROPER REGISTER
                                                        ; CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER 3 INDEX PULSES
002122  053151                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
002124  054742                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
002126  055204                          DF5             ;0,0,0,0
                                ;ITEM72
```

```
930  002130  052252                          EM72                    ;THERE WAS AN ERROR
931                                                                  ;AFTER A WRITE HEADER
932                                                                  ;AND DATA COMMAND
933
934  002132  054610                          DH72                    ;PC
935                                                                  ;RHCS1
936                                                                  ;RHCS2
937                                                                  ;RHDS1
938                                                                  ;RHDST
939                                                                  ;RHCA
940                                                                  ;RHER1
941                                                                  ;RHWC
942  002134  055146                          DT72                    ;$ERRPC,CS1,CS2,DS1,DST,CA,ER1,WC
943  002136  055273                          DF72                    ;0,0,0,0,0,0,0,0
944
945
946
947
948
949                              ;ITEM73
950  002140  052343                          EM73                    ;READING OVER 3 INDEX
951                                                                  ;PULSES CAUSED SC
952  002142  054610                          DH72
953  002144  055146                          DT72
954  002146  055273                          DF72
955
956                              ;ITEM74
957  002150  052513                          EM74                    ;READING OVER 3 INDEX
958                                                                  ;PULSES CAUSED OPI
959  002152  054610                          DH72
960  002154  055146                          DT72
961  002156  055273                          DF72
962
```

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


                                RPVEC:  254                             ;RP04 VECTOR ADDRESS
                                ;*****************************************************************
                                ;WORD COUNT REGISTER (RHWC)
                                ;EACH BIT IS CALLED BY BIT NUMBER


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


                                ;CONTROL AND STATUS REGISTER 2 (RHCS2)

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

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

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


                                ;CONTROL AND STATUS 1 REGISTER. (#00)

            000001              GO=     1                   ;GO (BIT #0)
            000100              IE=     100                 ;INTEPRUPT 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)
            000000              PSEL=   0                   ;PORT SELECT (BIT #10)
            004000              DVA=    4000                ;DEVICE AVAILABLE (BIT #11)
```

```
1019            020000                  MCPE=   20000                   ;MASSBUSS PARITY ERROR (BIT #13)
1020            040000                  TPE=    40000                   ;TRANSFER ERROR (BIT #14)
1021            100000                  SC=     100000                  ;SPECIAL CONDITION (BIT #15)
1022
1023                                    ;STATUS REGISTER (RHDS1) (#01)
1024
1025            000001                  DFF5=   1                       ;DRIVE FORWARD 5"/SEC.  (BIT #0)
1026            000002                  DFF20=  2                       ;DRIVE FORWARD 20"/SEC.  (BIT #1)
1027            000004                  DIGR=   4                       ;DRIVE TO INNER GAURD BAND (BIT #2)
1028            000010                  GRV=    10                      ;GO REVERSE (BIT #3)
1029            000020                  DL64=   20                      ;DIFFERENCE LESS THAN 64 (BIT #4)
1030            000040                  DE1=    40                      ;DIFFERENCE EQUALS 1 (BIT #5)
1031            000100                  VV=     100                     ;VOLUME VALID (BIT #6)
1032            000200                  DRY=    200                     ;DRIVE READY (BIT #7)
1033            000400                  DPR=    400                     ;DRIVE PRESENT (BIT #8)
1034            001000                  PROG=   1000                    ;PROGRAMABLE (BIT #9)
1035            002000                  LBT=    2000                    ;LAST SECTOR TRANSFERRED (BIT #10)
1036            004000                  WRL=    4000                    ;WRITE LOCK (BIT #11)
1037            010000                  MOL=    10000                   ;MEDIUM ON-LINE (BIT #12)
1038            020000                  PIP=    20000                   ;POSITIONING OPERATION IN PROGRESS (BIT #13)
1039            040000                  ERR=    40000                   ;COMPOSIT ERROR.  (BIT #14)
1040            100000                  ATA=    100000                  ;ATTENTION ACTIVE (BIT #15)
1041
1042                                    ;ERROR REGISTER #01 (RHER1) (#02)
1043            000001                  ILF=    1                       ;ILLEGAL FUNCTION (BIT #0)
1044            000002                  ILR=    2                       ;ILLEGAL REGISTER (BIT #1)
1045            000004                  RMR=    4                       ;REGISTER MODIFICATION REFUSED (BIT #2)
1046            000010                  PAR=    10                      ;PARITY ERROR (BIT #3)
1047            000020                  FER=    20                      ;FORMAT ERROR (BIT #4)
1048            000040                  WCF=    40                      ;WRITE CLOCK FAIL (BIT #5)
1049            000100                  ECH=    100                     ;ECC HARD ERROR (BIT #6)
1050            000200                  HCE=    200                     ;HEADER COMPARE ERROR (BIT #7)
1051            000400                  HCRC=   400                     ;HEADER CRC ERROR (BIT #8)
1052            001000                  AOE=    1000                    ;ADDRESS OVERFLOW ERROR (BIT #9)
1053            002000                  IAE=    2000                    ;INVALID ADDRESS ERROR (BIT #10)
1054            004000                  WLE=    4000                    ;WRITE LOCK ERROR (BIT #11)
1055            010000                  DTE=    10000                   ;DRIVE TIMING ERROR (BIT #12)
1056            020000                  OPI=    20000                   ;OPERATION INCOMPLETE (BIT #13)
1057            040000                  UNS=    40000                   ;DRIVE UNSAFE (BIT #14)
1058            100000                  DCK=    100000                  ;DATA CHECK ERROR (BIT 15)
1059
1060                                    ;MAINTAINABILITY REGISTER (RHMR)(#03)
1061
1062            000001                  DMD=    1                       ;DIAGINOSTIC MODE (BIT #0)
1063            000002                  MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
1064            000004                  MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
1065            000010                  MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1066            000020                  MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
1067            000040                  MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
1068            001000                  DTSY=   1000                    ;MAINTAINABILITY SYNC DETECTED (BIT #9)
1069
1070                                    ;ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1071
1072            000001                  AT0=    1                       ;DEVICE 0 (BIT #0)
1073            000002                  AT1=    2                       ;DEVICE 1 (BIT #1)
1074            000004                  AT2=    4                       ;DEVICE 2 (BIT #2)
```

```
1075    000010          AT3=    10              ;DEVICE 3 (BIT #3)
1076    000020          AT4=    20              ;DEVICE 4 (BIT #4)
1077    000040          AT5=    40              ;DEVICE 5 (BIT #5)
1078    000100          AT6=    100             ;DEVICE 6 (BIT #6)
1079    000200          AT7=    200             ;DEVICE 7 (BIT #7)
1080
1081                    ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1082                    ;EACH BIT IS CALLED BY BIT NUMBER
1083                    ;DRIVE TYPE REGISTER (RHDT) (#06)
1084                    ;EACH BIT IS CALLED BY BIT NUMBER
1085                    ;LOOK-AHEAD REGISTER (RHLA) (#07)
1086
1087    000001          EXT1=   1               ;EXTENSION 1 (BIT #0)
1088    000002          EXT2=   2               ;EXTENSION 2 (BIT #1)
1089    000004          EXT4=   4               ;EXTENSION 3 (BIT #2)
1090    000010          EXT10=  10              ;EXTENSION 4 (BIT #3)
1091    000020          EXT20=  20              ;EXTENSION 5 (BIT #4)
1092    000040          EXT40=  40              ;EXTENSION 6 (BIT #5)
1093    000100          SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
1094    000200          SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
1095    000400          SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
1096    001000          SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
1097    002000          SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
1098    004000          TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
1099    010000          TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
1100    020000          TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
1101    040000          TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
1102    100000          TRK20=  100000          ;TRACK FIELD 5 (BIT #15)
1103
1104                    ;ERROR REGISTER #2 (RHEP2) (#10)
1105
1106    000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1107    000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1108    000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1109    000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1110    000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
1111    000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1112    000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1113    000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
1114    000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1115    001000          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
1116    002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1117    004000          IXE=    4000            ;INDEX ERROR (BIT #11)
1118    010000          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
1119    020000          PLU=    20000           ;PLO UNSAFE (BIT #13)
1120    100000          ACU=    100000          ;ACUNSAFE (BIT #15)
1121
1122                    ;OFFSET REGISTER (RHOF) (#11)
1123
1124    000001          OF25=   1               ;OFFSET 25 MICRO INCHES (BIT #0)
1125    000002          OF50=   2               ;OFFSET 50 MICRO INCHES (BIT #1)
1126    000004          OF100=  4               ;OFFSET 100 MICRO INCHES (BIT #2)
1127    000010          OF200=  10              ;OFFSET 200 MICRO INCHES (BIT #3)
1128    000020          OF400=  20              ;OFFSET 400 MICRO INCHES (BIT #4)
1129    000040          OF800=  40              ;OFFSET 800 MICRO INCHES (BIT #5)
1130
```

```
1131            000200                  OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1132            002000                  HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
1133            004000                  ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1134            010000                  FMT22=  10000                   ;FORMAT BIT (BIT #12
1135
1136                                    ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
1137                                    ;EACH BIT IS CALLED BY BIT NUMBER.
1138
1139
1140
1141
1142                                    ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
1143                                    ;EACH BIT IS CALLED BY BIT NUMBER
1144
1145
1146
1147
1148                                    ;SERIAL NUMBER REGISTER (RHSN) (#14)
1149                                    ;EACH IS CALLED BY BIT NUMBER
1150
1151
1152
1153
1154                                    ;ERROR REGISTER #03 (RHER3) (#15)
1155
1156            000001                  PSU=    1                       ;PACK SPEED UNSAFE (BIT #0)
1157            000002                  VUF=    2                       ;VELOCITY UNSAFE (BIT #1)
1158            000010                  UWR=    10                      ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1159            000020                  PRE=    20                      ;DISK PACK ROTATION ERROR (BIT #4)
1160            000040                  ACL=    40                      ;AC LOW (BIT #5)
1161            000100                  DCL=    100                     ;DC LOW (BIT #6)
1162            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1163            100000                  OCYL=   100000                  ;OFF CYLINDER (BIT #15)
1164
1165
1166
1167                                    ;ECC POSITION REGISTER (RHEC1) (#16)
1168                                    ;EACH BIT IS CALLED BY BIT NUMBER
1169
1170
1171
1172
1173                                    ;ECC PATTERN REGISTER (RHEC2) (#17)
1174                                    ;EACH BIT IS CALLED BY BIT NUMBER
1175
1176
1177
1178
1179                                    ;*****************************************************************
1180
```

```
1181
1182
1183
1184                                    .SBTTL  REGISTER ADDRESSES
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196                                    ;RP04 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER
1197
1198  002162  176722                    RHDB:   176722                          ;DATA BUFFER
1199  002164  176702                    RHWC:   176702                          ;WORD COUNT
1200  002166  176704                    RHBA:   176704                          ;BUS ADDRESS
1201  002170  176710                    RHCS2:  176710                          ;CONTROL AND STATUS 2
1202
1203                                    ;RP04 DISK I/O REGISTERS LOCATED IN THE RP04 DEVICE LOGIC
1204
1205  002172  176700                    RHCS1:  176700                  ;CONTROL AND STATUS 1
1206  002174  176714                    RHER1:  176714                  ;ERROR #1
1207  002176  176706                    RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS
1208  002200  176740                    RHER2:  176740                  ;ERROR #2
1209  002202  176732                    RHOF:   176732                  ;OFFSET
1210  002204  176734                    RHCA:   176734                  ;DESIRED CYLINDER ADDRESS
1211  002206  176742                    RHER3:  176742                  ;ERROR #3
1212  002210  176716                    RHAS:   176716                  ;ATTENTION SUMMARY
1213  002212  176724                    RHMR:   176724                  ;MAINTAINABILITY
1214  002214  176712                    RHDS1:  176712                  ;DRIVE STATUS
1215  002216  176726                    RHDT:   176726                  ;DRIVE TYPE
1216  002220  176730                    RHSN:   176730                  ;SERIAL NUMBER
1217  002222  176744                    RHEC1:  176744                  ;ECC POSITION
1218  002224  176746                    RHEC2:  176746                  ;ECC PATTERN
1219  002226  176736                    PHCC:   176736                  ;CURRENT CYLINDER ADDRESS
1220  002230  176720                    RHLA:   176720                  ;LOOK-AHEAD
1221
1222
1223
1224
1225
1226                                    ;P-CLOCK (KW11-P) I/O REGISTERS
1227
1228  002232  172540                    PCLCSR: 172540                  ;CONTROL AND STATUS REGISTERS
1229  002234  172542                    PCLBUF: 172542                  ;COUNT SET BUFFER
1230  002236  172544                    PCLCTR: 172544                  ;COUNTER
1231
```

```
1232
1233                                ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTERS
1234                                ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1235                                ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1236                                ;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
1237
1238  002240  000000                DB:     0                       ;DATA BUFFER
1239  002242  000000                WC:     0                       ;WORD COUNT
1240  002244  000000                BA:     0                       ;BUS ADDRESS
1241  002246  000000                CS2:    0                       ;CONTROL AND STATUS 2
1242
1243
1244  002250  000000                CS1:    0                       ;CONTROL AND STATUS 1
1245  002252  000000                ER1:    0                       ;ERROR #1
1246  002254  000000                DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
1247  002256  000000                ER2:    0                       ;ERROR #2
1248  002260  000000                OF:     0                       ;OFFSET
1249  002262  000000                CA:     0                       ;DESIRED CYLINDER ADDRESS
1250  002264  000000                ER3:    0                       ;ERROR #3
1251  002266  000000                AS:     0                       ;ATTENTION SUMMARY
1252  002270  000000                MR:     0                       ;MAINTAINABILITY
1253  002272  000000                DS1:    0                       ;DRIVE STATUS
1254  002274  000000                DT:     0                       ;DRIVE TYPE
1255  002276  000000                SN:     0                       ;SERIAL NUMBER
1256  002300  000000                EC1:    0                       ;ECC POSITION
1257  002302  000000                EC2:    0                       ;ECC PATTERN
1258  002304  000000                CC:     0                       ;CURRENT CYLINDER ADDRESS
1259  002306  000000                LA:     0                       ;LOOK-AHEAD
1260
1261
1262
```

```
                                ;FUNCTION EQUATES
                                        ;TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
002310                          FUTABL:
002310  000000                  NOPERA: 0                       ;NO OPERATION
002312  000002                  UNLOAD: 2                       ;UNLOAD (STAND BY)
002314  000006                  RECALI: 6                       ;RECALIBRATE
002316  000010                  DCLEAR: 10                      ;DRIVE CLEAR
002320  000012                  RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
002322  000030                  SERCH:  30                      ;SEARCH COMMAND
002324  000050                  WRCHEK: 50                      ;WRITE CHECK DATA
002326  000052                  WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
002330  000060                  WRIDAT: 60                      ;WRITE DATA
002332  000062                  WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
002334  000070                  READAT: 70                      ;READ DATA
002336  000072                  REFOR:  72                      ;READ HEADER AND DATA
002340  000004                  SEECOM: 4                       ;SEEK COMMAND
002342  000014                  OFSETC: 14                      ;OFFSET COMMAND
002344  000016                  RETCL:  16                      ;RETURNTO CENTERLINE
002346  000022                  PKACK:  22                      ;PACK ACKNOWLEDGE
002350  000020                  READIN: 20                      ;READ IN
002352  000000                  ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION

                                        ;DATA BUFFER FOR READ WRITE
002354  000422                  WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
003420  000422                  REINTO: .BLKW   274.            ;READ INTO THIS BUFFER


                                ;RESERVED LOCATIONS
004464  000000                  REGADR: 0                       ;SAVE REGISTER ADDRESS HERE
004466  000000                  ERWORD: 0                       ;SAVE ERROR WORD NUMBER HERE
004470  000000                  TSTNM:  0                       ;TEST NUMBER
004472  000000                  RP4VEC: 0                       ;CONTAINS ADDRESS OF LOCATION
                                                                ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
                                                                ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
                                                                ;EFFECTIVE.

004474  000000                  OFSTVL: 0               ;OFFSET VALUE USED IN OFFSET TEST


004476  000024                  SAVERE: .BLKW   20.             ;BLOCK TO SAVE REGISTERS
004546  000000                  FINALA: 0                       ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
004550  000000                  FINACC: 0                       ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION


                                ;TABLE FOR ATTENTION BITS
                                ;ATTENTION TABLE
004552    001     002     004   ATABLE: .BYTE   1,2,4,10,20,40,100,200
004555    010     020     040
004560    100     200
```

```
1319                                    ;PESERVED LOCATIONS FOR UNIT SELECT
1320  004562  000010                    UNITS:  .BLKW   8.      ;THIS IS FILLED WITH -1
1321  004602  000000                    UNIT:   .WORD   0       ;UNIT UNDER TEST
1322  004604  000000                    NOUNIT: .WORD   0       ;NUMBER OF UNITS PRESENT
1323                                                            ;USED TO KEEP TRACK OF UNIT UNDER TEST
1324  004606  000000                    NUNIT:  .WORD   0       ;USED TO DETERMIN IF THERE ARE MORE
1325                                                            ;THAN ONE UNIT
1326  004610  000000                    NOPUSH: 0               ;ALL ONES INDICATE NONE OF THE OPERATOR
1327                                                            ;INTERVENTION TESTE WILL BE PERFORMED
1328  004612  000000                    SELECT: .WORD   0       ;ALL ONES INDICATE UNIT TO BE SELECTED
1329  004614  000000                    UNITSL: .WORD   0       ;UNIT NO. SELECTED
1330
1331
1332
1333  004616  000000                    ERFLG$: 0                       ;ERROR FLAG
1334  004620  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
1335                                                                            ;IF ONES WILL NOT TYPE HEADER
1336
1337
1338
1339  004622  000000                    ATTENT: 0               ;ATTENTION BIT FOR PRESENT UNIT
1340  004624  000000                    TOTALAT:        0                       ;TATAL ATTENTION BITS
1341
1342  004626  000000                    TMP0:   .WORD   0               ;TEMP STORAGE
1343  004630  000000                    TMP1:   .WORD   0
1344  004632  000000                    TMP4:   .WORD   0               ;TEMP STORAGE
1345  004634  000000                    TMP5:   .WORD   0               ;TEMP STORAGE
```

```
                                        .SBTTL  REGISTER TEST
004636  012737  177777  004610  BEGIN1: MOV     #-1,@#NOPUSH    ;JUMP OVER OPERATOR REQUIRED TESTS
004644  005037  004612                  CLR     @#SELECT        ;DO NOT SELECT UNIT
004650  000412                          BR      START
004652  012737  177777  004612  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
004660  005037  004610                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
004664  000404                          BR      START
004666  005037  004612          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
004672  005037  004610                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
                                                                ;NORMAL RUN

004676                          START:
004676  012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
004704  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
004710  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
004712  022706  001136                  CMP     #$TKS,R6        ;;DONE?
004716  001374                          BNE     .-6             ;;LOOP BACK IF NO
004720  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
004724  012737  033052  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
004732  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
004740  012737  034524  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
004746  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
004754  012737  036244  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
004762  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
004770  012737  036306  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
004776  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005004  005067  174174                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005010  005067  174172                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005014  112767  000001  174073          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005022  012767  005022  174056          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005030  012767  005030  174052          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS

005036  012767  000000  172732          MOV     #0,PS                   ;SET PROCESSOR STATUS TO 0
005044  012737  000200  000036          MOV     #200,@#TRAPVEC+2                ;TRAP PRIORITY = 4
005052  013700  002160                  MOV     @#RPVEC,R0              ;GET RP VECTOR ADDRESS
005056  012720  033010                  MOV     #RPVECT,(R0)+           ;THIS IS FOR UNTIMELY INTERRUPTS
005062  012710  000340                  MOV     #340,(R0)               ;RP04 INTERRUPT SERVICE ROUTINE
                                                                        ;PRIORITY = 7
005066  004737  033776                  JSR     PC,@#$TKINT     ;INITILIZE THE TK
005072  005737  004620                  TST     @#FIRST                 ;IS THIS FIRST TIME ROUND
005076  001001                          BNE     1$                      ;BRANCH IF NOT
005100  000402                          BR      2$
005102  000137  005706          1$:     JMP     @#SND1
005106                          2$:
005106  104400  005114                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
005112  000440                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/RP04 FUNCTIONAL CONTROLLER TEST PART II (STATIC 2B) - DER
005214                          64$:
005214  104400  005222                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
005220  000435                          BR      65$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/MAKE SURE PORT SWITCH ON DRIVE IS LOCKED ON EITHER PORT/
005314                          65$:
005314  104400  005322                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
005320  000440                          BR      66$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/IF CHANGE IS REQUIRED ON PORT SWITCH POWER DRIVE DOWN AND
```

```
1402  005422                          66$:
1403  005422  104400  005430                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1404  005426  000416                          BR      67$             ;;GET OVER THE ASCIZ
1405                                          ;;.ASCIZ        <15><12>/AGAIN AFTER SWITCH CHANGE/
1406  005464                          67$:
1407  005464  104400  005472                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1408  005470  000432                          BR      68$             ;;GET OVER THE ASCIZ
1409                                          ;;.ASCIZ        <15><12>/ALL DCL UNDER TEST MUST BE LOCKED ON CORRECT PORT/
1410  005556                          68$:
1411  005556  104400  005564                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1412  005562  000427                          BR      69$             ;;GET OVER THE ASCIZ
1413                                          ;;.ASCIZ        <15><12>/ALL DCL NOT UNDER TEST MUST BE SWITCHED OFF/
1414  005642                          69$:
1415  005642  104400  005650                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1416  005646  000417                          BR      70$             ;;GET OVER THE ASCIZ
1417                                          ;;.ASCIZ        <15><12>/OR LOCKED ON THE OTHER PORT/
1418  005706                          70$:
1419  005706  012737  177777  004620  SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER
1420
1421
1422                                  ;IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
1423                                  ;IF SO MAKE 'WAT" TRAPS GO TO 'WAIT.P'
1424                                  ;IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
1425                                  ;IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
1426                                  ;IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'
1427
1428                                  ;THE NEXT LINE IS TO BE ADDED LATER
1429                                  ;AND THE JUMP AND NOP REMOVED
1430                                  ;FOR NOW NO CLOCK WILL BE USED
1431                                  ;MOV@#1$,@#ERRVEC;SET TIME-OUT VECOTR
1432  005714  000137  005744                  JMP     @#1$            ;DO NOT USE CLOCK
1433  005720  000240                          NOP
1434  005722  005737  002232                  TST     @#PCLCSR        ;REFERENCE P-CLOCK STATUS REGISTER
1435                                                                   ;ADDRESS = 172540
1436  005726  012737  030312  036304          MOV     #WAIT.P,@#$TRPAD+20 ;THERE IS A P-CLOCK
1437  005734  012737  030252  004472          MOV     #TIME1,@#RP4VEC ;THERE IS A P CLOCK SO
1438                                                                   ;VECTOR TO TEME1
1439  005742  000406                          BR      2$
1440  005744  012737  030552  036304  1$:     MOV     #WAIT.T,@#$TRPAD+20 ;THERE IS NO P-CLOCK
1441  005752  012737  030264  004472          MOV     #TIME2,@#RP4VEC
1442  005760  012737  177777  034666  2$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
1443
1444
1445
1446  005766  005737  004612                  TST     @#SELECT                ;WAS IT A 200 START
1447
1448  005772  001442                          BEQ     TST1    ;BRANCH IF STARTING FROM 200
1449
1450  005774  104400  006002                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1451  006000  000424                          BR      64$             ;;GET OVER THE ASCIZ
1452                                          ;;.ASCIZ        <15><12>/SELECT UNIT NUMBER TO BE TESTED ?  /<15><12>
1453  006052                          64$:
1454  006052  104416                          RDOCT
1455  006054  042716  177770                  BIC     #177770,(SP)            ;ONLY KEEP LAST 3 BITS
1456  006060  011637  004602                  MOV     (SP),@#UNIT             ;SAVE UNIT TO BE TESTED
1457  006064  012637  004614                  MOV     (SP)+,@#UNITSL          ;SAVE UNIT TO BE TESTED
```

```
                                        BEQ     TST1    ;BRANCH IF STARTING FROM 200

006072  013737  004614  004602        MOV     @#UNITSL,@#UNIT         ;SET UNIT NUMBER

                        ;*********************************************************************
                        ;*TEST 1        REFERENCE EACH REGISTER
                        ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                        ;*********************************************************************
006100  000004          TST1:   SCOPE
006102  012767  000001  173074        MOV     #1,STIMES       ;;DO 1 ITERATION
006110  012706  001000                MOV     #STACK, SP      ;SET UP STACK POINTER
006114  012737  034532  000030        MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
                                                              ;NO REGISTERS ARE SAVED
006122  012737  006150  000004        MOV     #2$,    @#ERRVEC ;SET UP FOR BUS TIMEOUT
006130  012700  000024                MOV     #24,    R0      ;THERE ARE 24 REG TO TEST
006134  012701  002162                MOV     #RHDB,  R1      ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
006140  013102                1$:     MOV     @(R1)+, R2      ;READ HARDWARE REG.
006142  005300                        DEC     R0              ;COUNT DOWN
006144  001375                        BNE     1$              ;BRANCH IF 24 NOT DONE
006146  000471                        BR      3$              ;BRANCH IF 24 DONE
006150  012737  000006  000004  2$:    MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
006156  022626                        CMP     (SP)+,  (SP)+   ;CLEAN STACK
006160  016167  177776  173004        MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
006166  104007                        ERROR   7               ;REGISTER NON EXISTANT
006170  032737  020000  177570        BIT     #SW13,@#SWR     ;INHIBIT ERROR PRINTOUT ?
006176  001053                        BNE     4$              ;BRANCH IF YES
006200  104400  006206                TYPE    ,.+4            ;;TYPE ASCIZ STRING
006204  000431                        BR      64$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ      <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM /
006270                          64$:
006270  104400  006276                TYPE    ,.+4            ;;TYPE ASCIZ STRING
006274  000411                        BR      65$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ      <15><12>/AND RESTART AT /
006320                          65$:
006320  012746  032206                MOV     #BASECH,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
                                                              ;OF "CHANGE OF BASE ADDRESS" ROUTINE
006324  104402                        TYPOC
006326  000137  027544          4$:    JMP     @#$EOP          ;GO TO END OF PROGRAM
006332  012737  034524  000030  3$:    MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
                                                              ;SO THAT REGISTERS ARE SAVED
006340  012737  000006  000004        MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER

                        ;*********************************************************************
                        ;*TEST 2        PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
                        ;*      CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES
                        ;*********************************************************************
006346  000004          TST2:   SCOPE
006350  012767  000001  172626        MOV     #1,STIMES       ;;DO 1 ITERATION
006356  012706  001000                MOV     #STACK,SP       ;SET STACK POINTER
006362  013701  002210                MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
006366  012711  177777                MOV     #-1,@R1 ;THIS CLEARS RHAS (SURPRISED!)
006372  105711                        TSTB    @R1
```

```
1514
1515  006374  001407                          BEQ     TST3    ;BRANCH IF GOOD
1516
1517  006376  011137  001126                  MOV     @R1,@#BADDAT    ;BAD DATA
1518  006402  005037  001124                  CLR     @#GDDAT         ;GOOD DATA
1519  006406  010137  004464                  MOV     R1,@#REGADR     ;FAILING REG. RHAS
1520  006412  104005                          ERROR   5               ;RHAS DOES NOT CLEAR
1521                                                                   ;WITH ONES
1522
1523
1524                                  ;*****************************************************************
1525                                  ;*TEST 3        TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
1526                                  ;*      THE NUMBER OF RP04 DRIVES PRESENT ARE FOUND
1527                                  ;*      BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
1528                                  ;*      IN RHCS2 INCREMENTED FROM ZERO TO SEVEN
1529                                  ;*      THEN THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT
1530                                  ;*      THE DRIVE TYPE IS CHECKED TO HAVE 2020 OR 24020 AND THEN
1531                                  ;*      UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
1532
1533                                  ;*****************************************************************
1534  006414  000004                  TST3:   SCOPE
1535  006416  012767  000001  172560          MOV     #1,$TIMES       ;;DO 1 ITERATION
1536  006424  000005                          RESET           ;START WITH AN INIT
1537  006426  004737  033776                  JSP     PC,@#$TKINT     ;INITILIZE TK
1538  006432  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT?
1539  006440  001030                          BNE     4$              ;BRANCH IF YES
1540  006442  104400  006450                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1541  006446  000425                          BR      64$             ;;GET OVER THE ASCIZ
1542                                          ;;.ASCIZ        <15><12>/LOOKING AT RHAS -  RP04 DRIVES PRESENT /
1543  006522                          64$:
1544  006522  013701  002210          4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
1545  006526  013702  002170                  MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
1546  006532  005012                          CLR     @R2             ;CLEAR RHCS2
1547  006534  012700  000010                  MOV     #8.,R0          ;COUNT
1548  006540  013704  002174                  MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
1549  006544  012714  177777          1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
1550  006550  005212                          INC     @R2             ;INCREMENT UNIT NO.
1551  006552  005300                          DEC     R0              ;COUNT
1552  006554  001373                          BNE     1$              ;BRANCH IF 8 NOT DONE
1553  006556  111137  004624                  MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
1554                                                                   ;USED IN DRIVE CLEAR TEST
1555  006562  105037  004625                  CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
1556  006566  105711                          TSTB    @R1             ;TEST FOR ANY DRIVES PRESENT
1557  006570  001402                          BEQ     2$              ;IF SOME  NOT THERE BRANCH
1558  006572  000167  000420                  JMP     XE2             ;NONE THERE
1559  006576  032737  020000  177570  2$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPE OUT?
1560  006604  001402                          BEQ     3$              ;BRANCH IF NO
1561  006606  000167  000670                  JMP     TST4    ;OUT
1562  006612                          3$:
1563  006612  104400  006620                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1564  006616  000412                          BR      65$             ;;GET OVER THE ASCIZ
1565                                          ;;.ASCIZ        <15><12>/NO DRIVES-RHAS=0/
1566  006644                          65$:
1567  006644  104400  006652                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1568  006650  000436                          BR      66$             ;;GET OVER THE ASCIZ
1569                                          ;;.ASCIZ        <15><12>/WRITING ONES INTO ERROR REGISTER #1 FOR ALL UNIT NUMBERS/
```

```
1570  006746                          668$:
1571  006746  104400  006754                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1572  006752  000441                          BR      67$             ;;GET OVER THE ASCIZ
1573                                          ;;.ASCIZ      <15><12>/DOES NOT SET ANY BIT IN THE ATTENTION REGISTER SO ABORT P
1574  007056                          67$:
1575  007056  104400  007064                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1576  007062  000442                          BR      68$             ;;GET OVER THE ASCIZ
1577                                          ;;.ASCIZ      <15><12>/TO LOOP ON THIS TEST WITHOUT PRINTOUT SET SWITCHS 13  8
1578  007170                          68$:
1579  007170  000137  027544                  JMP     @#$EOP          ;GO OUT
1580  007174  005037  004562                  CLR     @#UNITS
1581  007200  012767  000001  175376          MOV     #1,NOUNIT       ;NO. UNITS PRESENT=1
1582  007206  005037  004602                  CLR     @#UNIT
1583  007212  000137  027544                  JMP     @#$EOP          ;JUMP OUT
1584  007216                          XE2:
1585  007216  012700  000010          2$:     MOV     #8.,R0          ;COUNTER
1586  007222  012703  004562                  MOV     #UNITS,R3       ;POINTER
1587  007226  012723  177777          3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
1588  007232  005300                          DEC     R0              ;COUNT
1589  007234  001374                          BNE     3$              ;BRANCH IF 8 NOT DONE
1590  007236  012703  004562                  MOV     #UNITS,R3       ;POINTER
1591  007242  005005                          CLR     R5
1592  007244  005037  004604                  CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
1593  007250  012700  000010                  MOV     #8.,R0          ;COUNTER
1594  007254  011137  001170                  MOV     @R1,@#TMP0      ;TEMPORARY STORAGE
1595  007260  006037  001170          4$:     ROR     @#TMP0 ;SET CARRY IF ONE IN  0 BIT
1596
1597  007264  103065                          BCC     5$
1598  007266  010577  172676                  MOV     R5,@RHCS2       ;INSERT UNIT NUMBER
1599  007272  022777  024020  172716          CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04
1600  007300  001450                          BEQ     6$              ;BRANCH IF YES
1601  007302  022777  020020  172706          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04
1602  007310  001444                          BEQ     6$              ;BRANCH IF YES
1603  007312  104400  007320                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1604  007316  000410                          BR      64$             ;;GET OVER THE ASCIZ
1605                                          ;;.ASCIZ      <15><12>/UNIT NUMBER /
1606  007340                          64$:
1607  007340  010546                          MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
1608  007342  104410                          TYPDS
1609  007344  104400  007352                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1610  007350  000405                          BR      65$             ;;GET OVER THE ASCIZ
1611                                          ;;.ASCIZ      /,  RHDT= /
1612  007364                          65$:
1613  007364  017746  172626                  MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
1614  007370  104402                          TYPOC
1615  007372  104400  007400                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1616  007376  000410                          BR      66$             ;;GET OVER THE ASCIZ
1617                                          ;;.ASCIZ      / ---NOT AN RP04/
1618  007420                          66$:
1619  007420  000407                          BR      5$              ;NO RP04 FOUND SO BRANCH
1620  007422  010523                  6$:     MOV     R5,(R3)+
1621  007424  104400  001215                  TYPE    ,$CRLF
1622  007430  010546                          MOV     R5,-(SP)
1623  007432  104410                          TYPDS                           ;TYPE DRIVE NO.
1624  007434  005237  004604                  INC     @#NOUNIT
1625  007440  005205                  5$:     INC     R5
```

```
1626  007442  005300                          DEC     R0
1627  007444  001305                          BNE     4$
1628  007446  013737  004562  004602          MOV     @#UNITS,@#UNIT
1629  007454  013737  004604  004606          MOV     @#NOUNIT,@#NUNIT        ;SAVE NO. OF UNITS
1630  007462  005337  004606                  DEC     @#NUNIT                 ;IF NUNIT = 0 THEN ONLY ONE UNIT
1631                                                                          ;IF NUNIT MORE THAN 0 THEN MORE THAN ONE UNIT
1632  007466  005737  004612                  TST     @#SELECT                ;STARTING ADDRESS 200 ?
1633
1634  007472  001403                          BEQ     TST4    ;BRANCH IF STARTING FROM 200
1635
1636  007474  013737  004614  004602          MOV     @#UNITSL,@#UNIT         ;SET UNIT NUMBER
1637
1638
1639
1640
1641
1642
1643
1644                                  ;*****************************************************************
1645                                  ;*TEST 4        TYPE SERIAL NUMBER AND DRIVE TYPE
1646                                  ;*      SET APPROPIATE ATTTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
1647                                  ;*      TYPE UNIT UNDER TEST
1648                                  ;*      READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
1649                                  ;*      TYPE IT OUT AND PROCEED
1650                                  ;*      TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
1651                                  ;*      SWITCHES 0 THRU 7 AND RESTART
1652
1653                                  ;*****************************************************************
1654  007502  000004                  TST4:   SCOPE
1655  007504  012767  000001  171472          MOV     #1,$TIMES       ;;DO 1 ITERATION
1656  007512  012767  007726  171366          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
1657  007520  012706  001000                  MOV     #STACK,SP       ;RESET STACK
1658  007524  012737  000004  004470          MOV     #4,@#TSTNM      ;SAVE TEST NUMBER
1659
1660  007532  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
1661                                                                  ;R3-RHDS1, R4-RHER1
1662                                                                  ;GIVE RH-11 INITIALIZE
1663                                                                  ;SETUP UNIT NUBER
1664  007536  005037  004622                  CLR     @#ATTENT        ;CLEAR
1665  007542  013700  004602                  MOV     @#UNIT,R0       ;R0 CONTAINS UNIT NO
1666  007546  116037  004552  004622          MOVB    ATABLE(R0),@#ATTENT ;SET APPROPIATE ATTENTION BIT
1667  007554  104400  007562                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1668  007560  000415                          BR      64$             ;;GET OVER THE ASCIZ
1669                                          ;;.ASCIZ      <15><12>/TESTING DRIVE NUMBER  /
1670  007614                          64$:
1671  007614  013746  004602                  MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
1672  007620  104402                          TYPOC                   ;TYPE  DRIVE NO.
1673  007622  104400  001215                  TYPE    ,$CRLF
1674  007626  104400  007634                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1675  007632  000410                          BR      65$             ;;GET OVER THE ASCIZ
1676                                          ;;.ASCIZ      <15><12>/SERIAL NO. = /
1677  007654                          65$:
1678  007654  017746  172340                  MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
1679  007660  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1680  007662  104400  001215                  TYPE    ,$CRLF
1681  007666  104400  007674                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
```

```
1682  007672  000410                          BR      66$             ;;GET OVER THE ASCIZ
1683                                          ;;.ASCIZ        <15><12>/DRIVE TYPE = /
1684  007714                          66$:
1685  007714  017746  172276                  MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
1686  007720  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1687  007722  104400  001215                  TYPE    ,$CRLF
1688  007726  005777  172266          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
1689  007732  005777  172260                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
1690  007736  017737  172256  002276          MOV     @RHSN,@$SN      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
1691  007744  017737  172246  002274          MOV     @RHDT,@$DT      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
1692
1693
1694
1695
1696                          ;*****************************************************************
1697                          ;*TEST 5         PROGRAM INTERRUPT
1698
1699                          ;*      PROGRAM INTERPUPT IS TESTED BY SETTING RDY AND IE
1700                          ;*      IN RHCS1 AT THE SAME TIME
1701                          ;*      THIS SHOULD INTERRUPT THROUGH LOCATION 254
1702                          ;*      THE PROCESSOR PRIORITY IS SET TO 4
1703                          ;*****************************************************************
1704  007752  000004          TST5:   SCOPE
1705
1706
1707  007754  012737  000005  004470          MOV     #6-1,@#TSTNM    ;THIS SAVES TEST NUMBER
1708
1709  007762  012706  001000                  MOV     #STACK,SP       ;RESET STACK
1710  007766  004737  030060                  JSR     PC,@#CLDISK     ;CLEAR DISK
1711  007772  013700  002160                  MOV     @#RPVEC,R0              ;GET RP VECTOR ADDRESS
1712  007776  012720  010040                  MOV     #RPTRP1,(R0)+           ;THIS IS FOR TIMELY INTERRUPTS
1713  010002  012710  000340                  MOV     #340,(R0)               ;RP04 INTERRUPT SERVICE ROUTINE
1714                                                                          ;PRIORITY = 7
1715
1716  010006  012767  000200  167762          MOV     #200,PS         ;SET PROCESSOP PRIORITY
1717  010014  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
1718  010020  013737  030550  001172          MOV     @#TIMCNT,@#$TMP1;COUNTER
1719  010026  005337  001172          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
1720  010032  001375                          BNE     1$              ;BRANCH IF NOT ZERO
1721                                                                  ;BEFORE THIS IS ZERO INTERRUPT SHOULD
1722                                                                  ;OCCUR
1723  010034  104065                          ERROR   65              ;INTERRUPT DID NOT OCCUR
1724
1725
1726  010036  000405                          BR      TST6    ;BRANCH TO NEXT TEST
1727
1728
1729  010040  022626          RPTRP1: CMP     (SP)+,(SP)+     ;RESTORE STACK
1730  010042  022711  004200                  CMP     #DVA!RDY,@R1    ;IE SHOULD BE LOW
1731
1732  010046  001401                          BEQ     TST6    ;BRANCH IF GOOD
1733
1734  010050  104065                          ERROR   65              ;INTERRUPT OCCURED BUT
1735                                                                  ;IE FAILED TO RESET
1736
1737
```

```
                        ;*****************************************************************
                        ;*TEST 6         INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
                        ;
                        ;*      PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
                        ;*      IE AND RDY IS SET, THIS SHOULD NOT INTERRUPT
                        ;*****************************************************************
010052  000004        TST6:   SCOPE

010054  012737  000006  004470          MOV     #7-1,@#TSTNM    ;THIS SAVES TEST NUMBER

010062  012706  001000          MOV     #STACK,SP       ;RESET STACK
010066  004737  030060          JSR     PC,@#CLDISK     ;CLEAR DISK
010072  013700  002160          MOV     @#RPVEC,R0              ;GET RP VECTOR ADDRESS
010076  012720  010136          MOV     #RPTRP2,(R0)+           ;THIS IS FOR UNTIMELY INTERRUPTS
010102  012710  000340          MOV     #340,(R0)               ;RP04 INTERRUPT SERVICE ROUTINE
                                                                ;PRIORITY = 7

010106  012767  000240  167662          MOV     #240,PS         ;SET PROCESSOR PRIORITY
010114  012711  000300          MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
010120  013737  030550  001172          MOV     @#TIMCNT,@#$TMP1;COUNTER
010126  005337  001172          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
010132  001375                  BNE     1$              ;BRANCH IF NOT ZERO
                                                        ;BEFORE THIS IS ZERO INTERRUPT SHOULD
                                                        ;OCCUR

010134  000402                  BR      TST7    ;NO INTERRUPT SO BRANCH


010136  022626          RPTRP2: CMP     (SP)+,(SP)+     ;RESTORE STACK
010140  104065                  ERROR   65              ;INTERRUPT OCCURRED WITH
                                                        ;PROCESSOR STATUS SAME
                                                        ;AS DISK


                        ;*****************************************************************
                        ;*TEST 7         SET VV BIT #6 IN RHDS1
                        ;*      THIS TEST SETS VV IN RHDS1 INCASE
                        ;*      ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
                        ;*      IS NOT PERFORMED
                        ;*      THERE IS A RESET AT THE BEGINING OF THIS TEST
                        ;*      FOR ERROR RECOVERY ONLY.

                        ;*****************************************************************
010142  000004        TST7:   SCOPE

                                ;IN CASE THERE IS ANY DRIVE EPRORS DURING POWER UP
                                ;OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
010144  000005                  RESET
010146  004737  033776          JSR     PC,@#$TKINT     ;INITILIZE TK
010152  012767  000000  167616          MOV     #0,PS
```

```
1794  010160  012706  001000                  MOV     #STACK,SP       ;RESET STACK
1795  010164  012737  000007  004470          MOV     #7,@#TSTNM      ;SAVE TEST NUMBER
1796
1797  010172  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
1798                                                                  ;R3-RHDS1, R4-RHER1
1799                                                                  ;GIVE RH-11 INITIALIZE
1800                                                                  ;SETUP UNIT NUBER
1801  010176  004737  030116                  JSR     PC,@#CHECK      ;CHECK DVA,RDY,MOL,DPP,DRY
1802
1803  010202  013777  002346  171762          MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
1804                                                                  ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
1805
1806
1807                                          ;NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
1808
1809  010210  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
1810  010214  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
1811  010216  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
1812                                                                  ;THE REGISTERS ARE SAVED
1813  010220  000022                          18.                     ;NUMBER OF REGISTERS
1814                                                                  ;SAVED = 18.
1815
1816  010222  013777  004472  171730          MOV     @#PP4VEC,@RPVEC ;SET PP04 VECTOR ADDRESS
1817                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
1818                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
1819                                                                  ;'TIME' WILL ONLY SAVE
1820                                                                  ;CURRENT CYLINDER ADDRESS
1821                                                                  ;AND LOOK AHEAD REGISTERS
1822
1823
1824  010230  013746  002346                  MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
1825  010234  052716  000001                  BIS     #GO,(SP)        ;GET READY TO SET GO
1826                                                                  ;WITHOUT INTERRUPT ENABLE
1827  010240  012677  171726                  MOV     (SP)+,@RHCS1    ;GO WITH
1828                                                                   ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
1829                                                                  ;WITH INTERRUPT DISABLED
1830
1831  010244  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
1832  010246  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
1833
1834
1835  010250  104420                          WAT                     ;WAIT FOR VV BIT TO SET
1836  010252  002214                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
1837  010254  000100                          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
1838  010256  000001                          1.                      ;ALLOW 10 MICRO SECONDS
1839  010260  000001                          1.                      ;VV MUST SET BETWEEN
1840                                                                  ;00 AND 20 MICRO SECONDS
1841                                          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
1842                                          ;R0 AND R5 IMMEDIATELY AFTER GO
1843  010262  013746  002346                  MOV     @#PKACK,-(SP)   ;SAVE COMMAND
1844  010266  052716  004200                  BIS     #DVA!RDY,(SP)   ;INCLUDE DVA!RDY
1845  010272  011637  001124                  MOV     (SP),@#GDDAT    ;SAVE FOR PRINTOUT
1846  010276  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!RDY
1847                                                                  ;AND COMMAND SHOULD BE SET
1848  010300  001405                          BEQ     64$             ;BRANCH IF GOOD
1849  010302  010037  001126                  MOV     R0,@#BDDAT      ;BAD DATA
```

```
1850  010306  010137  004464                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
1851  010312  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
1852                                                                ;COMMAND AND DVA!RDY SHOULD BE SET
1853  010314  012746  010700          64$:  MOV     #MOL!DPR!DRY!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
1854  010320  011637  001124                MOV     (SP),@#SGDDAT   ;SAVE FOR PRINTOUT
1855  010324  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!DPY!VV
1856                                                                ;SHOULD BE SET
1857  010326  001405                        BEQ     66$             ;BRANCH IF GOOD
1858  010330  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
1859  010334  010337  004464                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
1860  010340  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
1861                                                                ;MOL!DPR!DRY!VV SHOULD BE SET
1862  010342                          66$:
1863
1864  010342  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
1865  010346  002214                        RHDS1                   ;CHANGE RHDS1 REGISTER
1866
1867  010350  000001                        1                       ;1 BIT/BITS TO BE CHANGED
1868  010352  000001                        1                       ;NEW VALUE OF VV IS 1
1869  010354  000100                        VV                      ;CHANGE VV BIT
1870
1871                                        ;NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
1872                                        ;WITH AFTER PACK ACKNOWLEDGE
1873
1874
1875
1876  010356  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
1877                                                                ;PRESENT VALUE
1878  010362  004476                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
1879  010364  002242                        WC                      ;TEST DATA STARTING FROM 'RHWC'
1880  010366  000022                        18.                     ;18. REGISTERS TO BE COMPARED
1881  010370  010374                        1$                      ;RETURN TO 1$ ON ERROR
1882  010372  010400                        2$                      ;RETURN TO 2$ ON NO ERROR
1883
1884  010374  104015                  1$:   ERROR   15              ;GIVING A PACK ACKNOWLEDGE
1885  010376  000207                        RTS     PC              ;CAUSED AN ERROR
1886                                                                ;PACK ACKNOWLEDGE SHOULD
1887                                                                ;SET VV IN RHDS1
1888                                                                ;INTERRUPT SHOULD MAKE
1889                                                                ;IE = 0
1890                                                                ;NO OTHER REGISTERS SHOULD
1891                                                                ;CHANGE
1892                                                                ;GOOD DATA GIVES CONTENTS
1893                                                                ;OF REGISTER BEFORE COMMAND
1894                                                                ;RECEIVED DATA GIVES CONTENTS
1895                                                                ;OF REGISTER AFTER COMMAND
1896  010400                           2$:
1897
1898
1899
1900
1901
1902                                ;*****************************************************************
1903                                ;*TEST 10       LAST BLOCK TRANSFERED-RHDS1 BIT#10
1904
1905                                ;*      WRITE ONE WORD OF 65125 ON CYLINDER 410, TRACK 18
```

```
                                        ;*      SECTOR 21, BY A WRITE HEADER AND DATA COMMAND
                                        ;*      THEN CHECK ALL REGISTERS (LAST BLOCK TRANSFERED
                                        ;*      SHOULD BE SET)
                                        ;*      THEN READ ABOVE BY READ DATA 256 WORDS
                                        ;*      AGAIN LBT SHOULD BE SET
                                        ;*      CHECK ALL REGISTERS AND DATA
                                        ;*********************************************************************
010400  000004                    TST10:  SCOPE
010402  012706  001000                    MOV     #STACK,SP       ;RESET STACK
010406  012737  000010  004470            MOV     #10,@#TSTNM     ;SAVE TEST NUMBER
010414  004737  030060                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                        ;FILL WRITE FROM BUFFER WITH HEADER
010420  004037  027704                    JSP     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
010424  002354                    WRFROM                          ;LOCATION WHERE SAVED
010426  000005                    5                               ;NUMBER OF WORDS SAVED
010430  010632                    10632                           ;FIRST DATA WORD
010432  011025                    <18.*400>!<21.>                         ;SECOND DATA WORD
010434  000000                    0                               ;THIRD DATA WORD
010436  000000                    0                               ;FOURTH DATA WORD
010440  065125                    <26.*2000>!<18.*40>!<21.>                       ;FIFTH DATA WORD
                                        ;FILL READ INTO BUFFER WITH ALL ONES
010442  004037  027730                    JSR     R0,@#CLAREA     ;CLEAR 256 WORDS, FROM REINTO
010446  003420                    REINTO                          ;STARTING FROM REINTO
010450  000256                    256                             ;256 WORDS
010452  177777                    -1                              ;FILL WITH -1
                                        ;WRITE HEADER AND DATA IS LOADED
010454  004037  032024                    JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
010460  000632                    410.                            ;CYLINDER 410.
010462     025              .BYTE 21.                             ;SECTOR 21.
010463     022              .BYTE 18.                             ;TRACK 18.
010464  177773                    -1-4            ;WORD COUNT (DATA)=1+
                                                                ;4 HEADER WORDS
010466  002354                    WRFROM                          ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
010470  000000                    0                               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
010472  010000                    FMT22                           ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
010474  002332                    WRIFOR                          ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
                                        ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE
```

```
1962
1963  010476  004037  030220          JSR     R0,@#SAVER      ;SAVE REGISTERS
1964  010502  002164                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
1965  010504  004476                  SAVERE                  ;STARTING ADDRES OF WHERE
1966                                                          ;THE REGISTERS ARE SAVED
1967  010506  000022                  18.                     ;NUMBER OF REGISTERS
1968                                                          ;SAVED = 18.
1969
1970  010510  004737  030140          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DPY,VV
1971
1972  010514  013777  004472  171436  MOV     @#RP4VEC,@RPVEC ;SET PR#4 VECTOR ADDRESS
1973                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
1974                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
1975                                                          ;'TIME' WILL ONLY SAVE
1976                                                          ;CURRENT CYLINDER ADDRESS
1977                                                          ;AND LOOK AHEAD REGISTERS
1978
1979  010522  013746  002332          MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
1980  010526  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
1981                                                          ;ENABLE INTERRUPT
1982  010532  012677  171434          MOV     (SP)+,@RHCS1    ;GO WITH
1983                                                           ;62 IN RHCS1 FOR WRITE HEADER AND DATA
1984                                                          ;WITH INTERRUPT ENABLED
1985                                  ;TIME IS NOT CRITICAL
1986
1987  010536  104420                  WAT                     ;WAIT FOR LBT BIT TO SET
1988  010540  002214                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
1989  010542  002000                  LBT                     ;WAIT FOR LBT BIT IN RHDS1 REGISTER
1990  010544  004704                  2500.                   ;ALLOW 25000 MICRO SECONDS
1991  010546  004704                  2500.                   ;LBT MUST SET BETWEEN
1992                                                          ;00 AND 50000 MICRO SECONDS
1993
1994                                  ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
1995  010550  004037  027762          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
1996  010554  002164                  RHWC                    ;SAVED REGISTER TO CHANGE
1997  010556  000000                  0                       ;DATA
1998  010560  004037  027762          JSR     R0,@#FILLRE     ;MOV WRFROM+<5*2> INTO SAVED RHBA
1999  010564  002166                  RHBA                    ;SAVED REGISTER TO CHANGE
2000  010566  002366                  WRFROM+<5*2>                    ;DATA
2001
2002  010570  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2003  010574  002214                  RHDS1                   ;CHANGE RHDS1 REGISTER
2004
2005  010576  000001                  1                       ;1 BIT/BITS TO BE CHANGED
2006  010600  000001                  1                       ;NEW VALUE OF LBT IS 1
2007  010602  002000                  LBT                     ;CHANGE LBT BIT
2008
2009  010604  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2010  010610  002204                  RHCA                    ;CHANGE RHCA REGISTER
2011
2012  010612  000001                  1                       ;1 BIT/BITS TO BE CHANGED
2013  010614  000001                  1                       ;NEW VALUE OF BIT0 IS 1
2014  010616  000001                  BIT0                    ;CHANGE BIT0 BIT
2015  010620  004037  027762          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
2016  010624  002176                  RHDST                   ;SAVED REGISTER TO CHANGE
2017  010626  000000                  0                       ;DATA
```

```
2018
2019                                        ;COMPARE ALL REGISTERS
2020
2021  010630  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2022                                                                  ;PRESENT VALUE
2023  010634  004476                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
2024  010636  002242                        WC                      ;TEST DATA STARTING FROM 'RHWC'
2025  010640  000021                        17.                     ;17. REGISTERS TO BE COMPARED
2026  010642  010646                        1$                      ;RETURN TO 1$ ON ERROR
2027  010644  010652                        2$                      ;RETURN TO 2$ ON NO ERROR
2028
2029  010646  104045              1$:       ERROP   45              ;WRITING ON THE LAST BLOCK
2030  010650  000207                        RTS     PC              ;IE. CYLINDER 410, SECTOR 21
2031                                                                  ;TRACK 18 CAUSED
2032                                                                  ;IMPROPER REGISTER CHANGE
2033                                                                  ;GOOD DATA GIVES WHAT
2034                                                                  ;SHOULD BE THERE
2035                                                                  ;RECEIVED DATA GIVES WHAT
2036                                                                  ;WAS THERE AFTER WRITE
2037                                                                  ;ON LAST BLOCK
2038
2039                                        ;NOW A READ DATA WILL BE DONE ON SAME CYLINDER, SECTOR
2040                                        ;TRACK
2041                                        ;CLEAR ERRORS
2042  010652                        2$:
2043
2044  010652  004737  030060                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2045                                                                  ;R3-RHDS1, R4-RHER1
2046                                                                  ;GIVE RH-11 INITIALIZE
2047                                                                  ;SETUP UNIT NUBER
2048
2049                                        ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
2050
2051  010656  004037  027704                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
2052  010662  002354                        WRFROM                  ;LOCATION WHERE SAVED
2053  010664  000001                        1                       ;NUMBER OF WORDS SAVED
2054  010666  065125                        <26.*2000>!<18.*40>!<21.>                       ;FIRST DATA WORD
2055  010670  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+2
2056  010674  002356                        WRFROM+2                        ;STARTING FROM WRFROM+2
2057  010676  000400                        256.                    ;256. WORDS
2058  010700  000000                        0                       ;FILL WITH 0
2059
2060
2061                                        ;READ COMMAND IS LOADED
2062
2063  010702  004037  032024                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
2064  010706  000632                        410.                    ;CYLINDER 410.
2065  010710     025                .BYTE   21.                     ;SECTOR 21.
2066  010711     022                .BYTE   18.                     ;TRACK 18.
2067  010712  177400                        -256.                   ;WORD COUNT = 256.
2068  010714  003420                        REINTO                  ;BUS ADDRESS
2069                                                                  ;STARTING ADDRESS OF DATA
2070                                                                  ;BUFFER = REINTO
2071  010716  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2072  010720  010000                        FMT22                   ;16 BITS PER WORD FORMAT
2073                                                                  ;DO NOT INHIBIT ECC CORRECTION
```

```
2074                                                    ;DO NOT INHIBIT HEADER COMPARE
2075  010722  002334                    READAT          ;GET READY TO DO A READAT
2076                                                ;READ DATA WITH 70 IN RHCS1
2077
2078
2079                                    ;SAVE REGISTERS FOR COMPARISON AFTER READ DATA
2080  010724  004037  030220            JSR     R0,@#SAVER      ;SAVE REGISTERS
2081  010730  002164                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2082  010732  004476                    SAVERE                  ;STARTING ADDRES OF WHERE
2083                                                            ;THE REGISTERS ARE SAVED
2084  010734  000022                    18.                     ;NUMBER OF REGISTERS
2085                                                            ;SAVED = 18.
2086  010736  004737  030140            JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2087
2088  010742  013777  004472  171210    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2089                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
2090                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2091                                                            ;'TIME' WILL ONLY SAVE
2092                                                            ;CURRENT CYLINDER ADDRESS
2093                                                            ;AND LOOK AHEAD REGISTERS
2094
2095  010750  013746  002334            MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
2096  010754  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2097                                                            ;ENABLE INTERRUPT
2098  010760  012677  171206            MOV     (SP)+,@RHCS1    ;GO WITH
2099                                                             ;70 IN RHCS1 FOR READ DATA
2100                                                            ;WITH INTERRUPT ENABLED
2101
2102                                    ;TIME IS NOT CRITICAL
2103
2104  010764  104420                    WAT                     ;WAIT FOR RDY BIT TO SET
2105  010766  002172                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
2106  010770  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2107  010772  001614                    900.                    ;ALLOW 9000 MICRO SECONDS
2108  010774  001502                    834.                    ;RDY MUST SET BETWEEN
2109                                                            ;740 AND 17420 MICRO SECONDS
2110                                    ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
2111  010776  004037  027762            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
2112  011002  002164                    RHWC                    ;SAVED REGISTER TO CHANGE
2113  011004  000000                    0                       ;DATA
2114  011006  004037  027762            JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
2115  011012  002166                    RHBA                    ;SAVED REGISTER TO CHANGE
2116  011014  004420                    REINTO+<256.*2>                 ;DATA
2117
2118  011016  004037  030732            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2119  011022  002214                    RHDS1                   ;CHANGE RHDS1 REGISTER
2120
2121  011024  000001                    1                       ;1 BIT/BITS TO BE CHANGED
2122  011026  000001                    1                       ;NEW VALUE OF LBT IS 1
2123  011030  002000                    LBT                     ;CHANGE LBT BIT
2124
2125  011032  004037  030732            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2126  011036  002204                    RHCA                    ;CHANGE RHCA REGISTER
2127
2128  011040  000001                    1                       ;1 BIT/BITS TO BE CHANGED
2129  011042  000001                    1                       ;NEW VALUE OF BIT0 IS 1
```

```
2130  011044  000001                          BIT0                    ;CHANGE BIT0 BIT
2131  011046  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
2132  011052  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
2133  011054  000000                          0                       ;DATA
2134
2135                                          ;COMPARE ALL REGISTERS
2136
2137  011056  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2138                                                                  ;PRESENT VALUE
2139  011062  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
2140  011064  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
2141  011066  000022                          18.                     ;18. REGISTERS TO BE COMPARED
2142  011070  011074                          3$                      ;RETURN TO 3$ ON ERROR
2143  011072  011100                          4$                      ;RETURN TO 4$ ON NO ERROR
2144
2145  011074  104045                  3$:     ERROR   45              ;READING ON LAST BLOCK IE.
2146  011076  000207                          RTS     PC              ;CYLINDER 10, SECTOR 21, TRACK 18
2147                                                                  ;CAUSED AN ERROR
2148                                                                  ;GOOD DATA GIVES WHAT SHOULD
2149                                                                  ;BE THERE
2150                                                                  ;RECEIVED DATA GIVES WHAT
2151                                                                  ;WAS THERE AFTER READ
2152                                                                  ;FROM LAST BLOCK
2153                                          ;READ DATA WILL BE COMPARED
2154  011100                          4$:
2155
2156  011100  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
2157  011104  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
2158  011106  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
2159  011110  000400                          256.                    ;256. WORDS TO BE COMPARED
2160  011112  011116                          5$                      ;RETURN TO 5$ ON ERROR
2161  011114  011122                          6$                      ;RETURN TO 6$ ON NO ERROR
2162
2163
2164  011116  104046                  5$:     ERROR   46              ;DATA READ FROM
2165  011120  000207                          RTS     PC              ;LAST BLOCK IN ERROR
2166
2167  011122                          6$:
2168
2169
2170
2171
2172                                          ;*************************************************************************
2173                                          ;*TEST 11        SEAPCH COMMAND
2174
2175                                          ;*      THE SEARCH COMMAND WILL BE DONE ON CYLINDER 0
2176                                          ;*      THAT IS STARTING WITH A RECALIBRATE
2177                                          ;*      THEN HEADER AND DATA WILL BE WRITTEN FOR SECTOR 0 AND 1
2178                                          ;*      ALL REGISTERS WILL BE CHECKED
2179                                          ;*      A SEARCH COMMAND WILL BE GIVEN FOR SECTOR 0
2180                                          ;*      ON INTERRUPT SECTOR 1 HEADER AND DATA WILL BE READ
2181                                          ;*      TIME WILL BE CRITICAL AS THE TIME TAKEN TO DO THE
2182                                          ;*      READ IS THE ONLY INDICATOR THAT THE HEADS WERE ON
2183                                          ;*      SECTOR 0 AT INTERRUPT TIME.  TIME ALLOWED IS MAXIMUM
2184                                          ;*      OF 1500 MICRO SECONDS
2185                                          ;*      THEN ALL REGISTERS ARE CHECKED AND DATA READ
```

```
                                ;*      IS  CHECKED
                                ;*****************************************************************
011122  000004                TST11:  SCOPE
011124  012706  001000                MOV     #STACK,SP       ;RESET STACK
011130  012737  000011  004470        MOV     #11,@#TSTNM     ;SAVE TEST NUMBER
011136  004737  030060                JSP     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUBER
                                        ;GET HEADS TO CYLINDER 0
011142  004737  030140                JSP     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
011146  013777  004472  171004        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                              ;'TIME' WILL ONLY SAVE
                                                              ;CURRENT CYLINDER ADDRESS
                                                              ;AND LOOK AHEAD REGISTERS
011154  013746  002314                MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
011160  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                              ;ENABLE INTERRUPT
011164  012677  171002                MOV     (SP)+,@RHCS1    ;GO WITH
                                                               ;6 IN RHCS1 FOR RECALIBRATE
                                                              ;WITH INTERRUPT ENABLED
011170  104420                        WAT                     ;WAIT FOR RDY BIT TO SET
011172  002214                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
011174  000200                        RDY                     ;WAIT FOR RDY BIT IN RHDS1 REGISTER
011176  012740                        5600.                   ;ALLOW 56000 MICRO SECONDS
011200  012737                        5599.                   ;RDY MUST SET BETWEEN
                                                              ;10 AND 111990 MICRO SECONDS
011202  004737  030060                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUBER
                                        ;FILL WRITE FROM BUFFER WITH HEADER
011206  004037  027704                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
011212  002354                        WRFROM                  ;LOCATION WHERE SAVED
011214  000004                        4                       ;NUMBER OF WORDS SAVED
011216  010000                        10000                   ;FIRST DATA WORD
011220  000000                        0                       ;SECOND DATA WORD
011222  000000                        0                       ;THIRD DATA WORD
011224  000000                        0                       ;FOURTH DATA WORD
                                        ;FILL WRITE FROM BUFFER WITH DATA
011226  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
011232  002364                        WRFROM+10                       ;STARTING FROM WRFROM+10
011234  000400                        256.                    ;256. WORDS
011236  000000                        0                       ;FILL WITH 0
```

```
                                        ;FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER

011240  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
011244  003364                          WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
011246  000004                          4                       ;NUMBER OF WORDS SAVED
011250  010000                          10000                   ;FIRST DATA WORD
011252  000001                          1                       ;SECOND DATA WORD
011254  000000                          0                       ;THIRD DATA WORD
011256  000000                          0                       ;FOURTH DATA WORD

                                        ;FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
011260  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<264.*2>
011264  003374                          WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
011266  000004                          4                       ;4 WORDS
011270  000001                          1                       ;FILL WITH 1

                                        ;CLEAR READ INTO BUFFER WITH DATA OTHER THAN EXPECTED DATA
011272  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
011276  003420                          REINTO                  ;STARTING FROM REINTO
011300  000404                          260.                    ;260. WORDS
011302  000377                          377                     ;FILL WITH 377

                                        ;THE WRITE HEADER AND DATA WILL BE LOADED

011304  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
011310  000000                          0                       ;CYLINDER 0
011312    000                   .BYTE   0                       ;SECTOR 0
011313    000                   .BYTE   0                       ;TRACK 0
011314  177364                          -264.-4         ;WORD COUNT (DATA)=264.+
                                                                ;4 HEADER WORDS
011316  002354                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
011320  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
011322  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
011324  002332                          WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1

                                        ;SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
011326  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
011332  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
011334  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
011336  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

011340  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV

011344  013777  004472  170606          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
```

```
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

011352  013746  002332                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
011356  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
011362  012677  170604                MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                        ;WITH INTERRUPT ENABLED
011366  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
011370  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

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

011372  104420                        WAT                     ;WAIT FOR RDY BIT TO SET
011374  002172                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
011376  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
011400  001614                        900.                    ;ALLOW 9000 MICRO SECONDS
011402  001507                        839.                    ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS

                                      ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                      ;R0 AND R5 IMMEDIATELY AFTER GO
011404  013746  002332                MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
011410  052716  004101                BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
011414  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
011420  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
                                                        ;AND COMMAND SHOULD BE SET
011422  001405                        BEQ     67$             ;BRANCH IF GOOD
011424  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
011430  010137  004464                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
011434  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                        ;COMMAND AND IE!DVA!GO SHOULD BE SET
011436  012746  010500        67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
011442  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
011446  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
011450  001405                        BEQ     69$             ;BRANCH IF GOOD
011452  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
011456  010337  004464                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
011462  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
011464                          69$:

                                      ;NOW CHANGE SAVE REGISTERS TO EXPECTED VALUES
011464  004037  027762                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
011470  002164                        RHWC                    ;SAVED REGISTER TO CHANGE
011472  000000                        0                       ;DATA
011474  004037  027762                JSR     R0,@#FILLRE     ;MOV WRFROM+<268.*2> INTO SAVED RHBA
011500  002166                        RHBA                    ;SAVED REGISTER TO CHANGE
011502  003404                        WRFROM+<268.*2>                 ;DATA
011504  004037  027762                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
011510  002176                        RHDST                   ;SAVED REGISTER TO CHANGE
011512  000002                        2                       ;DATA
```

```
                                        ;NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
                                        ;WITH REGISTERS AFTER COMMAND

011514  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
011520  004476                          SAVERF                  ;GOOD DATA SAVED IN 'SAVERF'
011522  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
011524  000022                          18.                     ;18. REGISTERS TO BE COMPARED
011526  011532                          1$                      ;RETURN TO 1$ ON ERROR
011530  011536                          2$                      ;RETURN TO 2$ ON NO ERROR

011532  104027                  1$:     ERROR   27              ;WRITE HEADER AND DATA
011534  000207                          RTS     PC              ;CAUSED IMPROPER REGISTER
                                                                ;CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER COMMAND

                                        ;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
011536                          2$:

011536  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER

011542  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
011546  000000                          0                       ;CYLINDER 0
011550     000                  .BYTE   0                       ;SECTOR 0
011551     000                  .BYTE   0                       ;TRACK 0
011552  177770                          -8.                     ;WORD COUNT = 8.
011554  003420                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
011556  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
011560  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
011562  002322                          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
011564  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
011570  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
```

```
2410  011572  004476                          SAVERF                  ;STARTING ADDRES OF WHERE
2411                                                                  ;THE REGISTERS ARE SAVED
2412  011574  000022                          18.                     ;NUMBER OF REGISTERS
2413                                                                  ;SAVED = 18.
2414
2415                                          ;NOW SAVE VALUES FOR RHCS1 AND RHDST WHICH
2416                                          ;WILL CHANGE AFTER THE SEARCH
2417  011576  013746  002336                  MOV     @#REFOR,-(SP)   ;SAVE READ HEADER AND DATA
2418  011602  052716  000101                  BIS     #IE!GO,(SP)     ;INTERRUPT ENABLE AND GO
2419  011606  012637  004626                  MOV     (SP)+,@#TMP0    ;SAVE IN R0 FOR RHCS1
2420  011612  012737  000001  004634          MOV     #1,@#TMP5       ;SAVE TRACK 0 SECTOR 1 FOR RHDST
2421
2422                                          ;THE INTERRUPT VECTOR WILL BE SET TO GO TO 2#
2423                                          ;AFTER THE SEARCH
2424  011620  012777  011660  170332          MOV     #7#,@RPVEC      ;SET INTERRUPT VECTOR TO 2#
2425  011626  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2426
2427
2428  011632  013746  002322                  MOV     @#SERCH,-(SP)   ;GET READY TO MOVE COMMAND
2429  011636  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2430                                                                  ;ENABLE INTERRUPT
2431  011642  012677  170324                  MOV     (SP)+,@RHCS1    ;GO WITH
2432                                                                  ;WITH INTERRUPT ENABLED
2433
2434                                          ;TIME IS NOT CRITICAL THIS ONLY WAITS FOR SEARCH COMPLETION
2435
2436  011646  104420                          WAT                     ;WAIT FOR DRY BIT TO SET
2437  011650  002214                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
2438  011652  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
2439  011654  001614                          9#8.                    ;ALLOW 9#8# MICRO SECONDS
2440  011656  001507                          839.                    ;DRY MUST SET BETWEEN
2441                                                                  ;69# AND 17479 MICRO SECONDS
2442
2443  011660  012767  000000  166110  78:     MOV     #0,PS           ;SET PROSESSOR STATUS TO
2444                                                                  ;PRIORITY 0 IN CASE IT WAS
2445                                                                  ;TAKEN OUT OF WAT ROUTINE
2446                                                                  ;BEFORE RTI
2447  011666  013777  004634  170302          MOV     @#TMP5,@RHDST   ;SET DESIRED SECTOR/TRACK
2448                                                                  ;REGISTER TO SECTOR 1,TRACK 0
2449  011674  013777  004472  170256          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2450                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
2451                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2452                                                                  ;'TIME' WILL ONLY SAVE
2453                                                                  ;CURRENT CYLINDER ADDRESS
2454                                                                  ;AND LOOK AHEAD REGISTERS
2455
2456  011702  013777  004626  170262          MOV     @#TMP0,@RHCS1   ;FILL RHCS1 WITH READ COMMAND
2457                                                                  ;TOGETHER WITH INTERRUPT ENABLE
2458                                                                  ;AND GO
2459
2460                                          ;TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
2461                                          ;INDICATES WRONG SEARCH IN THE SEARCH COMMAND
2462
2463  011710  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
2464  011712  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
2465  011714  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
2466  011716  000225                        149.                    ;ALLOW 1490 MICRO SECONDS
2467  011720  000002                        2.                      ;RDY MUST SET BETWEEN
2468                                                                  ;1470 AND 1510 MICRO SECONDS
2469
2470                                        ;WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
2471
2472  011722  004037  027704                JSP     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
2473  011726  002354                        WRFROM                  ;LOCATION WHERE SAVED
2474  011730  000004                        4                       ;NUMBER OF WORDS SAVED
2475  011732  010000                        10000                   ;FIRST DATA WORD
2476  011734  000001                        1                       ;SECOND DATA WORD
2477  011736  000000                        0                       ;THIRD DATA WORD
2478  011740  000000                        0                       ;FOURTH DATA WORD
2479  011742  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
2480  011746  002364                        WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
2481  011750  000004                        4                       ;4 WORDS
2482  011752  000001                        1                       ;FILL WITH 1
2483
2484
2485                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
2486  011754  004037  027762                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
2487  011760  002164                        RHWC                    ;SAVED REGISTER TO CHANGE
2488  011762  000000                        0                       ;DATA
2489  011764  004037  027762                JSR     R0,@#FILLRE     ;MOV REINTO+<8.*2> INTO SAVED RHBA
2490  011770  002166                        RHBA                    ;SAVED REGISTER TO CHANGE
2491  011772  003440                        REINTO+<8.*2>                   ;DATA
2492  011774  004037  027762                JSR     R0,@#FILLRE     ;MOV 4272 INTO SAVED RHCS1
2493  012000  002172                        RHCS1                   ;SAVED REGISTER TO CHANGE
2494  012002  004272                        4272                    ;DATA
2495  012004  004037  027762                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
2496  012010  002176                        RHDST                   ;SAVED REGISTER TO CHANGE
2497  012012  000002                        2                       ;DATA
2498
2499                                        ;COMPARE REGISTER BEFORE READ HEADER AND DATA
2500                                        ;WITH REGISTERS AFTER COMMAND
2501
2502
2503  012014  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2504                                                                  ;PRESENT VALUE
2505  012020  004476                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
2506  012022  002242                        WC                      ;TEST DATA STARTING FROM 'RHWC'
2507  012024  000022                        18.                     ;18. REGISTERS TO BE COMPARED
2508  012026  012032                        3$                      ;RETURN TO 3$ ON ERROR
2509  012030  012036                        4$                      ;RETURN TO 4$ ON NO ERROR
2510  012032  104031              3$:       ERROR   31              ;READ HEADER AND DATA CAUSED
2511  012034  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
2512                                                                  ;GOOD DATA GIVES WHAT SHOULD
2513                                                                  ;BE THERE
2514                                                                  ;RECEIVED DATA GIVES WHAT WAS
2515                                                                  ;THERE AFTER COMMAND
2516
2517                                        ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE
2518                                        ;THE READ WAS GOOD
2519  012036                        4$:
2520
2521  012036  004037  032070                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
```

```
2522  012042  002354                        WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
2523  012044  003420                        REINTO                  ;TEST DATA STARTS FROM REINTO
2524  012046  000010                        8.                      ;8. WORDS TO BE COMPARED
2525  012050  012054                        5$                      ;RETURN TO 5$ ON ERROR
2526  012052  012060                        6$                      ;RETURN TO 6$ ON NO ERROR
2527
2528
2529  012054  104053              5$:       ERROR   53              ;READ HEADER AND DATA
2530  012056  000207                        RTS     PC              ;AFTER A SEARCH CAUSED
2531                                                                ;AN ERROR
2532  012060                      6$:
2533
2534
2535                              ;*****************************************************************
2536                              ;*TEST 12       SEARCH COMMAND
2537
2538                              ;*      THE ONLY THING NEW IN THIS TEST IS AN IMPLIED SEEK
2539                              ;*      IN A SEARCH COMMAND
2540                              ;*      THE HEADS START FROM CYLINDER 10 BY A SEEK
2541                              ;*      COMMAND THEN A SEARCH SECTOR 0 TRACK 0 CYLINDER 0
2542                              ;*      IS GIVEN
2543                              ;*      THEN A READ COMMAND IS GIVEN FOR
2544                              ;*      CYLINDER 0, TRACK 0, SECTOR 1
2545                              ;*      TIME FOR THE READ IS THE ONLY INDICATOR
2546                              ;*      OF CORRECT SEARCH
2547
2548                              ;*****************************************************************
2549  012060  000004              TST12:  SCOPE
2550  012062  012706  001000              MOV     #STACK,SP       ;RESET STACK
2551  012066  012737  000012  004470      MOV     #12,@#TSTNM     ;SAVE TEST NUMBER
2552
2553  012074  004737  030060              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2554                                                              ;R3-RHDS1, R4-RHER1
2555                                                              ;GIVE RH-11 INITIALIZE
2556                                                              ;SETUP UNIT NUBER
2557
2558                              ;GET THE HEADS TO CYLINDER 10
2559  012100  004737  030140              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2560
2561  012104  013777  004472  170046      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2562                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
2563                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2564                                                              ;'TIME' WILL ONLY SAVE
2565                                                              ;CURRENT CYLINDER ADDRESS
2566                                                              ;AND LOOK AHEAD REGISTERS
2567
2568  012112  004037  030030              JSR     R0,@#SEEKCY     ;SEEK FOR
2569  012116  000012                      10.                     ;CYLINDER 10.
2570
2571  012120  013746  002340              MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
2572  012124  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2573                                                              ;ENABLE INTERRUPT
2574  012130  012677  170036              MOV     (SP)+,@RHCS1    ;GO WITH
2575                                                               ;4 IN RHCS1 FOR SEEK
2576                                                              ;WITH INTERRUPT ENABLED
2577
```

```
2578  012134  104420                    WAT                     ;WAIT FOR DRY BIT TO SET
2579  012136  002214                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
2580  012140  000200                    DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
2581  012142  015530                    7000.                   ;ALLOW 70000 MICRO SECONDS
2582  012144  000043                    35.                     ;DRY MUST SET BETWEEN
2583                                                            ;69650 AND 70350 MICRO SECONDS
2584
2585                                    ;FILL REGISTERS FOR READ HEADER AND DATA TO BE DONE AFTER SEARCH
2586
2587  012146  004737  030060            JSR     PC,@#CLDISK     ;SET R1=RHCS1, R2=RHCS2
2588                                                            ;R3=RHDS1, R4=RHER1
2589                                                            ;GIVE RH-11 INITIALIZE
2590                                                            ;SETUP UNIT NUBER
2591
2592  012152  004037  032024            JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
2593  012156  000000                    0                       ;CYLINDER 0
2594  012160     000             .BYTE  0                       ;SECTOR 0
2595  012161     000             .BYTE  0                       ;TRACK 0
2596  012162  177770                    -8.                     ;WORD COUNT = 8.
2597  012164  003420                    REINTO                  ;BUS ADDRESS
2598                                                            ;STARTING ADDRESS OF DATA
2599                                                            ;BUFFER = REINTO
2600  012166  000000                    0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2601  012170  014000                    ECI!FMT22                       ;16 BITS PER WORD FORMAT
2602                                                            ;INHIBIT ECC CORRECTION
2603                                                            ;DO NOT INHIBIT HEADER COMPARE
2604  012172  002322                    SERCH                   ;GET READY TO DO A SERCH
2605                                                            ;SEARCH WITH 30 IN RHCS1
2606
2607
2608                                    ;SAVE REGISTERS FOR COMPARISON AFTER SEARCH
2609                                    ;AND READ HEADER AND DATA
2610  012174  004037  030220            JSR     R0,@#SAVER      ;SAVE REGISTERS
2611  012200  002164                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2612  012202  004476                    SAVERE                  ;STARTING ADDRES OF WHERE
2613                                                            ;THE REGISTERS ARE SAVED
2614  012204  000022                    18.                     ;NUMBER OF REGISTERS
2615                                                            ;SAVED = 18.
2616
2617                                    ;NOW GIVE THE SEARCH COMMAND
2618  012206  004737  030140            JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2619
2620  012212  012777  012376  167740    MOV     #38,@PVEC       ;INTERRUPT VECTOR SET TO 38
2621
2622  012220  004037  030010            JSR     R0,@#SRCH       ;SEARCH FOR
2623  012224  000000                    0                       ;CYLINDER 0
2624  012226     000             .BYTE  0                       ;SECTOR 0
2625  012227     000             .BYTE  0                       ;TRACK 0
2626
2627  012230  013700  002322            MOV     @#SERCH,R0      ;EXPECTED CONTENTS OF RHCS1
2628                                                            ;IMMEDIATELY AFTER GO
2629  012234  052700  004301            BIS     #DVA!RDY!IE!GO,R0       ;EXPECTED BITS IN RHCS1
2630  012240  012705  010500            MOV     #MOL!DPR!VV,R5 ;EXPECTED BITS IN RHDS1
2631                                                            ;IMMEDIATELY AFTER GO
2632
2633
```

```
2634  012244  013746  002322                MOV     @#SERCH,-(SP)   ;GET READY TO MOVE COMMAND
2635  012250  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2636                                                                  ;ENABLE INTERRUPT
2637  012254  012677  167712                MOV     (SP)+,@RHCS1    ;GO WITH
2638                                                                  ;WITH INTERRUPT ENABLED
2639  012260  021100                        CMP     @R1,P0          ;IS RHCS1 GOOD
2640  012262  001413                        BEQ     1$              ;BRANCH IF GOOD
2641  012264  011137  001126                MOV     @R1,@#BDDAT     ;BAD DATA FOR RHCS1
2642  012270  010037  001124                MOV     R0,@#GDDAT      ;GOOD DATA
2643  012274  010137  004464                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
2644  012300  012737  000340  000036        MOV     #340,@#TRAPVEC+2                ;TRAP PRIORITY = 7
2645  012306  104021                        ERROR   21              ;DURING SEARCH COMMAND
2646                                                                  ;CONTENTS OF RHCS1 WAS
2647                                                                  ;NOT AS EXPECTED
2648  012310  000414                        BR      2$              ;IF LAST ERROR 21 OCCURRED
2649                                                                  ;THEN DO NOT CHECK RHDS1
2650                                                                  ;AS TOO MUCH TIME HAS
2651                                                                  ;PASSED
2652
2653  012312  021305                1$:     CMP     @R3,R5          ;IS RHDS1 GOOD
2654  012314  001412                        BEQ     2$              ;BRANCH IF GOOD
2655  012316  011337  001126                MOV     @R3,@#BDDAT     ;BAD DATA FOR RHDS1
2656  012322  010537  001124                MOV     R5,@#GDDAT      ;GOOD DATA
2657  012326  010337  004464                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
2658  012332  012737  000340  000036        MOV     #340,@#TRAPVEC+2                ;TRAP PRIORITY = 7
2659  012340  104063                        ERROR   63              ;DURING SEARCH COMMAND
2660                                                                  ;CONTENTS OF RHDS1 WAS
2661                                                                  ;IN CORRECT
2662
2663  012342  013737  002336  004626  2$:    MOV     @#REFOR,@#TMP0  ;SAVE READ HEADER AND DATA
2664  012350  052737  000101  004626        BIS     #IE!GO,@#TMP0   ;INCLUDE INTERRUPT ENABLE, GO
2665  012356  012737  000001  004634        MOV     #1,@#TMP5       ;SAVE TRACK 0, SECTOR 1
2666
2667                                        ;THIS IS ONLY A WAIT LOOP
2668
2669  012364  104420                        WAT                     ;WAIT FOR RDY BIT TO SET
2670  012366  002214                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
2671  012370  000200                        RDY                     ;WAIT FOR RDY BIT IN RHDS1 REGISTER
2672  012372  015530                        7000.                   ;ALLOW 70000 MICRO SECONDS
2673  012374  000043                        35.                     ;RDY MUST SET BETWEEN
2674                                                                  ;69650 AND 70350 MICRO SECONDS
2675
2676  012376  012737  000200  000036  3$:    MOV     #200,@#TRAPVEC+2                ;TRAP PRIORITY = 4
2677  012404  012767  000000  165364        MOV     #0,PS           ;SET PROSESSOR STATUS TO 0
2678  012412  013777  004634  167556        MOV     @#TMP5,@RHDST   ;SET DESIRED SECTOR/TRACK
2679                                                                  ;REGISTER TO SECTOR 1, TRACK 0
2680
2681  012420  013777  004472  167532        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2682                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
2683                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2684                                                                  ;'TIME' WILL ONLY SAVE
2685                                                                  ;CURRENT CYLINDER ADDRESS
2686                                                                  ;AND LOOK AHEAD REGISTERS
2687  012426  013711  004626                MOV     @#TMP0,@R1              ;FILL RHCS1 WITH READ COMMAND
2688                                                                  ;TOGETHER WITH INTERRUPT ENABLE
2689                                                                  ;AND GO
```

```
2690
2691                                    ;TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
2692                                    ;INDICATES WRONG SEARCH IN THE SEARCH COMMAND
2693
2694  012432  104420                    WAT                     ;WAIT FOR RDY BIT TO SET
2695  012434  002172                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
2696  012436  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2697  012440  000225                    149.                    ;ALLOW 1490 MICRO SECONDS
2698  012442  000002                    2.                      ;RDY MUST SET BETWEEN
2699                                                            ;1470 AND 1510 MICRO SECONDS
2700
2701                                    ;WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
2702
2703  012444  004037  027704            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
2704  012450  002354                    WRFROM                  ;LOCATION WHERE SAVED
2705  012452  000004                    4                       ;NUMBER OF WORDS SAVED
2706  012454  010000                    10000                   ;FIRST DATA WORD
2707  012456  000001                    1                       ;SECOND DATA WORD
2708  012460  000000                    0                       ;THIRD DATA WORD
2709  012462  000000                    0                       ;FOURTH DATA WORD
2710  012464  004037  027730            JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
2711  012470  002364                    WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
2712  012472  000004                    4                       ;4 WORDS
2713  012474  000001                    1                       ;FILL WITH 1
2714
2715
2716                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
2717  012476  004037  027762            JSR     R0,@#FILLRF     ;MOV 0 INTO SAVED PHWC
2718  012502  002164                    RHWC                    ;SAVED REGISTER TO CHANGE
2719  012504  000000                    0                       ;DATA
2720  012506  004037  027762            JSR     R0,@#FILLRE     ;MOV REINTO+<8.*2> INTO SAVED RHBA
2721  012512  002166                    RHBA                    ;SAVED REGISTER TO CHANGE
2722  012514  003440                    REINTO+<8.*2>                   ;DATA
2723  012516  004037  027762            JSR     R0,@#FILLRE     ;MOV 4272 INTO SAVED RHCS1
2724  012522  002172                    RHCS1                   ;SAVED REGISTER TO CHANGE
2725  012524  004272                    4272                    ;DATA
2726  012526  004037  027762            JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
2727  012532  002176                    RHDST                   ;SAVED REGISTER TO CHANGE
2728  012534  000002                    2                       ;DATA
2729  012536  004037  027762            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
2730  012542  002226                    RHCC                    ;SAVED REGISTER TO CHANGE
2731  012544  000000                    0                       ;DATA
2732
2733                                    ;COMPARE REGISTER BEFORE READ HEADER AND DATA
2734                                    ;WITH REGISTERS AFTER COMMAND
2735
2736
2737  012546  004037  031040            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2738                                                            ;PRESENT VALUE
2739  012552  004476                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
2740  012554  002242                    WC                      ;TEST DATA STARTING FROM 'PHWC'
2741  012556  000022                    18.                     ;18. REGISTERS TO BE COMPARED
2742  012560  012564                    4$                      ;RETURN TO 4$ ON ERROR
2743  012562  012570                    5$                      ;RETURN TO 5$ ON NO ERROR
2744
2745  012564  104031            4$:     ERROR   31              ;READ HEADER AND DATA CAUSED
```

```
2746  012566  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
2747                                                                  ;GOOD DATA GIVES WHAT SHOULD
2748                                                                  ;BE THERE
2749                                                                  ;RECEIVED DATA GIVES WHAT WAS
2750                                                                  ;THERE AFTER COMMAND
2751
2752                                          ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE
2753                                          ;THE READ WAS GOOD
2754  012570                          5$:
2755
2756  012570  004037  032070                  JSR     R0,@$COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
2757  012574  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
2758  012576  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
2759  012600  000010                          8.                      ;8. WORDS TO BE COMPARED
2760  012602  012606                          6$                      ;RETURN TO 6$ ON ERROR
2761  012604  012612                          7$                      ;RETURN TO 7$ ON NO ERROR
2762
2763
2764  012606  104053                  6$:     ERROR   53              ;READ HEADER AND DATA
2765  012610  000207                          RTS     PC              ;AFTER A SEARCH CAUSED
2766                                                                  ;AN ERROR
2767  012612                          7$:
2768
2769
2770
2771
2772                                  ;*      THE NEXT TEST REMOVES SECTOR 1 ON CYLINDER 0
2773                                  ;*      TRACK0 AND PUTS SECTOR 0 THERE.
2774                                  ;*      HENCE THE PACK IS UNFORMATTED FROM
2775                                  ;*      THIS POINT ON TO THE TEST WHEN SECTOR
2776                                  ;*      1 IS REPLACED.  IF TESTING IS STOPPED WITH
2777                                  ;*      AN ERROR IN THE SECTION OF THE PROGRAM BETWEEN
2778                                  ;*      THIS AND WHEN SECTOR 1 IS REPLACED THEN THE
2779                                  ;*      DISK BEING USED MAY HAVE BEEN UNFORMATTED
2780                                  ;*      IF THE LAST PASS OF THIS PROGRAM GIVES
2781                                  ;*      NO ERRORS IN THIS SECTION THEN THE DISK
2782                                  ;*      MAY NOT HAVE BEEN UNFORMATTED, HOWEVER IT
2783                                  ;*      IS RECOMMENDED THAT AFTER A PASS OF THIS
2784                                  ;*      PROGRAM THE DISK BE REFORMATTED.
2785
2786
2787
2788
2789
2790                                  ;*********************************************************************
2791                                  ;*TEST 13       HEADER COMPARE ERROR - RHER1 BIT #7
2792
2793                                  ;*      WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
2794                                  ;*      AND PUT SECTOR 0 THERE ON CYLINDER 0
2795                                  ;*      THEN A READ DATA IS GIVEN FOR SECTOR1
2796                                  ;*      HCE- BIT #7 IN RHER1 SHOULD SET.
2797                                  ;*      ALL REGISTERS ARE CHECKED
2798                                  ;*      ANY DATA READ IS CHECKED
2799
2800                                  ;*********************************************************************
2801  012612  000004                  TST13:  SCOPE
```

```
2802  012614  012706  001000                  MOV     #STACK,SP       ;RESET STACK
2803  012620  012737  000013  004470          MOV     #13,@#TSTNM     ;SAVE TEST NUMBER
2804
2805  012626  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2806                                                                  ;R3-RHDS1, R4-RHER1
2807                                                                  ;GIVE RH-11 INITIALIZE
2808                                                                  ;SETUP UNIT NUBER
2809
2810                                          ;FILL WRITE FROM BUFFER WITH HEADER
2811
2812  012632  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
2813  012636  002354                          WRFROM                  ;LOCATION WHERE SAVED
2814  012640  000005                          5                       ;NUMBER OF WORDS SAVED
2815  012642  010000                          10000                   ;FIRST DATA WORD
2816  012644  000000                          0                       ;SECOND DATA WORD
2817  012646  000000                          0                       ;THIRD DATA WORD
2818  012650  000000                          0                       ;FOURTH DATA WORD
2819  012652  000001                          1                       ;FIFTH DATA WORD
2820
2821                                          ;FILL READ INTO BUFFER WITH ALL ONES
2822
2823  012654  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
2824  012660  003420                          REINTO                  ;STARTING FROM REINTO
2825  012662  000400                          256.                    ;256. WORDS
2826  012664  177777                          -1                      ;FILL WITH -1
2827
2828
2829                                          ;WRITE HEADER AND DATA IS LOADED
2830
2831  012666  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
2832  012672  000000                          0.                      ;CYLINDER 0.
2833  012674     001                  .BYTE   1.                      ;SECTOR 1.
2834  012675     000                  .BYTE   0.                      ;TRACK 0.
2835  012676  177773                          -1-4            ;WORD COUNT (DATA)=1+
2836                                                                  ;4 HEADER WORDS
2837  012700  002354                          WRFROM                  ;BUS ADDRESS
2838                                                                  ;STARTING ADDRESS OF DATA
2839                                                                  ;BUFFER = WRFROM
2840  012702  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2841  012704  010000                          FMT22                   ;16 BITS PER WORD FORMAT
2842                                                                  ;DO NOT INHIBIT ECC CORRECTION
2843                                                                  ;DO NOT INHIBIT HEADER COMPARE
2844  012706  002332                          WRIFOP                  ;GET READY TO DO A WRIFOR
2845                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
2846
2847
2848                                          ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE
2849  012710  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
2850  012714  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2851  012716  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
2852                                                                  ;THE REGISTERS ARE SAVED
2853  012720  000021                          17.                     ;NUMBER OF REGISTERS
2854                                                                  ;SAVED = 17.
2855  012722  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2856
2857  012726  013777  004472  167224          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
```

```
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

012734  013746  002332                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
012740  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
012744  012677  167222                MOV     (SP)+,@PHCS1    ;GO WITH
                                                         ;67 IN RHCS1 FOR WRITE HEADER AND DATA
                                                        ;WITH INTERRUPT ENABLED
                                      ;TIME IS NOT CRITICAL

012750  104420                       WAT                     ;WAIT FOR RDY BIT TO SET
012752  002172                       RHCS1                   ;WAIT FOR RHCS1 REGISTER
012754  000200                       RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
012756  004704                       2500.                   ;ALLOW 25000 MICRO SECONDS
012760  004704                       2500.                   ;RDY MUST SET BETWEEN
                                                        ;00 AND 50000 MICRO SECONDS
                                      ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
012762  004037  027762                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
012766  002164                       RHWC                    ;SAVED REGISTER TO CHANGE
012770  000000                       0                       ;DATA
012772  004037  027762                JSR     R0,@#FILLRF     ;MOV WRFROM+<5*2> INTO SAVED RHBA
012776  002166                       RHBA                    ;SAVED REGISTER TO CHANGE
013000  002366                       WRFROM+<5*2>                    ;DATA
013002  004037  027762                JSR     P0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
013006  002176                       RHDST                   ;SAVED REGISTER TO CHANGE
013010  000002                       2                       ;DATA
                                      ;COMPARE ALL REGISTERS

013012  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
013016  004476                       SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
013020  002242                       WC                      ;TEST DATA STARTING FROM 'RHWC'
013022  000021                       17.                     ;17. REGISTERS TO BE COMPARED
013024  013030                       18                      ;RETURN TO 18 ON ERROR
013026  013034                       28                      ;RETURN TO 28 ON NO ERROR
013030  104027                18:    ERROR   27              ;WRITING HEADER AND DATA CAUSED
013032  000207                       RTS     PC
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT
                                                        ;SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER WRITE

                                      ;NOW A READ DATA WILL BE DONE ON CYLINDER=0, SECTOR=1,
                                      ;TRACK=0
                                      ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
013034                          28:

013034  004737  030060                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUBER
```

```
2914  013040  004037  027730                  JSP     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
2915  013044  002354                          WRFROM                  ;STARTING FROM WRFROM
2916  013046  000400                          256.                    ;256. WORDS
2917  013050  177777                          -1                      ;FILL WITH -1
2918
2919                                          ;READ COMMAND IS LOADED
2920
2921  013052  004037  032024                  JSP     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
2922  013056  000000                          0                       ;CYLINDER 0
2923  013060     001                  .BYTE   1.                      ;SECTOR 1.
2924  013061     000                  .BYTE   0                       ;TRACK 0
2925  013062  177777                          -1                      ;WORD COUNT = 1
2926  013064  003420                          REINTO                  ;BUS ADDRESS
2927                                                                  ;STARTING ADDRESS OF DATA
2928                                                                  ;BUFFER = REINTO
2929  013066  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
2930  013070  010000                          FMT22                   ;16 BITS PER WORD FORMAT
2931                                                                  ;DO NOT INHIBIT ECC CORRECTION
2932                                                                  ;DO NOT INHIBIT HEADER COMPARE
2933  013072  002334                          READAT                  ;GET READY TO DO A READAT
2934                                                        ;READ DATA WITH 70 IN RHCS1
2935
2936                                          ;SAVE REGISTERS FOR COMPARISON AFTER READ DATA
2937  013074  004037  030220                  JSP     R0,@#SAVER      ;SAVE REGISTERS
2938  013100  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2939  013102  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
2940                                                                  ;THE REGISTERS ARE SAVED
2941  013104  000022                          18.                     ;NUMBER OF REGISTERS
2942                                                                  ;SAVED = 18.
2943  013106  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
2944
2945  013112  013777  004472  167040          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2946                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
2947                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2948                                                                  ;'TIME' WILL ONLY SAVE
2949                                                                  ;CURRENT CYLINDER ADDRESS
2950                                                                  ;AND LOOK AHEAD REGISTERS
2951
2952  013120  013746  002334                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
2953  013124  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2954                                                                  ;ENABLE INTERRUPT
2955  013130  012677  167036                  MOV     (SP)+,@RHCS1    ;GO WITH
2956                                                                   ;70 IN RHCS1 FOR READ DATA
2957                                                                  ;WITH INTERRUPT ENABLED
2958                                          ;TIME IS NOT CRITICAL
2959
2960  013134  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
2961  013136  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
2962  013140  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2963  013142  010110                          4168.                   ;ALLOW 41680 MICRO SECONDS
2964  013144  001502                          834.                    ;RDY MUST SET BETWEEN
2965                                                                  ;33340 AND 50020 MICRO SECONDS
2966                                          ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
2967
2968  013146  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2969  013152  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
```

```
2970
2971  013154  000002                          2                       ;2 BIT/BITS TO BE CHANGED
2972  013156  000001                          1                       ;NEW VALUE OF ATA IS 1
2973  013160  100000                          ATA                     ;CHANGE ATA BIT
2974  013162  000001                          1                       ;NEW VALUE OF ERR IS 1
2975  013164  040000                          ERR                     ;CHANGE ERR BIT
2976
2977  013166  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2978  013172  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
2979
2980  013174  000002                          2                       ;2 BIT/BITS TO BE CHANGED
2981  013176  000001                          1                       ;NEW VALUE OF SC IS 1
2982  013200  100000                          SC                      ;CHANGE SC BIT
2983  013202  000001                          1                       ;NEW VALUE OF TRE IS 1
2984  013204  040000                          TRE                     ;CHANGE TRE BIT
2985
2986  013206  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2987  013212  002174                          RHER1                   ;CHANGE RHER1 REGISTER
2988
2989  013214  000001                          1                       ;1 BIT/BITS TO BE CHANGED
2990  013216  000001                          1                       ;NEW VALUE OF HCE IS 1
2991  013220  000200                          HCE                     ;CHANGE HCE BIT
2992  013222  004037  027762                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
2993  013226  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
2994  013230  000002                          2                       ;DATA
2995
2996  013232  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
2997                                                                  ;FOR WORKING DRIVE IN
2998                                                                  ;SAVED RHAS LOACTION
2999
3000                                  ;COMPARE ALL REGISTERS
3001
3002  013240  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3003                                                                  ;PRESENT VALUE
3004  013244  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3005  013246  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3006  013250  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3007  013252  013256                          3$                      ;RETURN TO 3$ ON ERROR
3008  013254  013262                          4$                      ;RETURN TO 4$ ON NO ERROR
3009  013256  104047                  3$:     ERROR   47              ;READING ON NON EXISTANT SECTOR
3010  013260  000207                          RTS     PC
3011                                                                  ;CAUSED AN ERROR
3012                                                                  ;GOOD DATA GIVES WHAT SHOULD
3013                                                                  ;BE THERE
3014                                                                  ;RECEIVED DATA GIVES WHAT
3015                                                                  ;WAS THERE AFTER READ
3016                                  ;READ DATA WILL BE COMPARED
3017  013262                  4$:
3018
3019  013262  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
3020  013266  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
3021  013270  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
3022  013272  000400                          256.                    ;256. WORDS TO BE COMPARED
3023  013274  013300                          5$                      ;RETURN TO 5$ ON ERROR
3024  013276  013304                          6$                      ;RETURN TO 6$ ON NO ERROR
3025
```

```
3026  013300  104050                  5$:     ERROR   50              ;DATA READ FROM NON
3027  013302  000207                          RTS     PC              ;EXISTANT SECTOR CAUSED AN ERROR
3028  013304                          6$:
3029
3030                                  ;*****************************************************************
3031                                  ;*TEST 14       HEADER COMPARE ERROR - RHER1 BIT #7
3032
3033                                  ;*      WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
3034                                  ;*      AND PUT SECTOR 0 ON CYLINDER 0
3035                                  ;*      THEN A WRITE DATA IS GIVEN FOR SECTOR 1, TRACK 0, CYLINDER 0
3036                                  ;*      FOR 70. WORDS
3037                                  ;*      HCE - BIT 7 IN RHER1 SHOULD SET
3038                                  ;*      ALL REGISTERS ARE CHECKED
3039                                  ;*      THEN A READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
3040                                  ;*      IS GIVEN, HCE - BIT 7 SHOULD SET AND ALL
3041                                  ;*      HEADER AND DATA SHOULD BE READ
3042
3043                                  ;*****************************************************************
3044  013304  000004                  TST14:  SCOPE
3045  013306  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3046  013312  012737  000014  004470          MOV     #14,@#TSTNM     ;SAVE TEST NUMBER
3047
3048  013320  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3049                                                                  ;R3-RHDS1, R4-RHER1
3050                                                                  ;GIVE RH-11 INITIALIZE
3051                                                                  ;SETUP UNIT NUBER
3052
3053                                          ;FILL WRITE FROM BUFFER WITH HEADER ADD DATA
3054
3055  013324  004037  027704                  JSP     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
3056  013330  002354                          WRFROM                  ;LOCATION WHERE SAVED
3057  013332  000006                          6                       ;NUMBER OF WORDS SAVED
3058  013334  010000                          10000                   ;FIRST DATA WORD
3059  013336  000000                          0                       ;SECOND DATA WORD
3060  013340  000000                          0                       ;THIRD DATA WORD
3061  013342  000000                          0                       ;FOURTH DATA WORD
3062  013344  000001                          1                       ;FIFTH DATA WORD
3063  013346  000001                          1                       ;SIXTH DATA WORD
3064
3065                                          ;FILL READ INTO BUFFER WITH ALL ONES
3066  013350  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
3067  013354  003420                          REINTO                  ;STARTING FROM REINTO
3068  013356  000400                          256.                    ;256. WORDS
3069  013360  177777                          -1                      ;FILL WITH -1
3070
3071
3072                                          ;WRITE HEADER AND DATA IS LOADED
3073
3074  013362  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3075  013366  000000                          0                       ;CYLINDER 0
3076  013370     001                  .BYTE   1                       ;SECTOR 1
3077  013371     000                  .BYTE   0                       ;TRACK 0
3078  013372  177772                          -2-4            ;WORD COUNT (DATA)=2+
3079                                                                  ;4 HEADER WORDS
3080  013374  002354                          WRFROM                  ;BUS ADDRESS
3081                                                                  ;STARTING ADDRESS OF DATA
```

```
                                                        ;BUFFER = WRFROM
013376  000000                      0                 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
013400  010000                      FMT22             ;16 BITS PER WORD FORMAT
                                                      ;DO NOT INHIBIT ECC CORRECTION
                                                      ;DO NOT INHIBIT HEADER COMPARE
013402  002332                      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
013404  004037  030220                JSR     R0,@#SAVER        ;SAVE REGISTERS
013410  002164                       RHWC                      ;RHWC IS THE FIRST REGISTER SAVED
013412  004476                       SAVERE                    ;STARTING ADDRES OF WHERE
                                                               ;THE REGISTERS ARE SAVED
013414  000021                       17.                       ;NUMBER OF REGISTERS
                                                               ;SAVED = 17.
013416  004737  030140                JSR     PC,@#CHECKT       ;CHECK DVA,RDY,MOL,DPR,DRY,VV

013422  013777  004472  166530        MOV     @#PP4VEC,@RPVEC   ;SET PP04 VECTOR ADDRESS
                                                               ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                               ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                               ;'TIME' WILL ONLY SAVE
                                                               ;CURRENT CYLINDER ADDRESS
                                                               ;AND LOOK AHEAD REGISTERS

013430  013746  002332                MOV     @#WRIFOR,-(SP)    ;GET READY TO MOVE COMMAND
013434  052716  000101                BIS     #GO!IE,(SP)       ;GET READY TO SET GO AND
                                                               ;ENABLE INTERRUPT
013440  012677  166526                MOV     (SP)+,@RHCS1      ;GO WITH
                                                                ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                               ;WITH INTERRUPT ENABLED
                                                ;TIME IS NOT CRITICAL

013444  104420                       WAT                       ;WAIT FOR RDY BIT TO SET
013446  002172                       RHCS1                     ;WAIT FOR RHCS1 REGISTER
013450  000200                       RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
013452  004704                       2500.                     ;ALLOW 25000 MICRO SECONDS
013454  004704                       2500.                     ;RDY MUST SET BETWEEN
                                                               ;00 AND 50000 MICRO SECONDS

                                                ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
013456  004037  027762                JSR     P0,@#FILLRE       ;MOV 0 INTO SAVED RHWC
013462  002164                       RHWC                      ;SAVED REGISTER TO CHANGE
013464  000000                       0                         ;DATA
013466  004037  027762                JSR     R0,@#FILLRE       ;MOV WRFROM+<6*2> INTO SAVED RHBA
013472  002166                       RHBA                      ;SAVED REGISTER TO CHANGE
013474  002370                       WRFROM+<6*2>                      ;DATA
013476  004037  027762                JSR     R0,@#FILLRE       ;MOV 2 INTO SAVED RHDST
013502  002176                       RHDST                     ;SAVED REGISTER TO CHANGE
013504  000002                       2                         ;DATA
                                                ;COMPARE ALL REGISTERS

013506  004037  031040                JSR     R0,@#COMREG       ;COMPARE SAVED REGISTERS WITH
                                                               ;PRESENT VALUE
013512  004476                       SAVERE                    ;GOOD DATA SAVED IN 'SAVERE'
013514  002242                       WC                        ;TEST DATA STARTING FROM 'RHWC'
```

```
3138  013516  000021                          17.                     ;17. REGISTERS TO BE COMPARED
3139  013520  013524                          1$                      ;RETURN TO 1$ ON ERROR
3140  013522  013530                          2$                      ;RETURN TO 2$ ON NO ERROR
3141  013524  104027                  1$:     ERROR   27              ;WRITING HEADER AND DATA CAUSED
3142  013526  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
3143                                                                  ;GOOD DATA GIVES WHAT
3144                                                                  ;SHOULD BE THERE
3145                                                                  ;RECEIVED DATA GIVES WHAT
3146                                                                  ;WAS THERE AFTER WRITE
3147
3148                                          ;NOW A WRITE DATA WILL BE DONE ON SAME CYLINDER, SECTOR
3149                                          ;TRACK
3150
3151  013530                          2$:
3152
3153                                          ;FILL WRITE FROM BUFFER WITH DATA
3154  013530  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 70. WORDS, FROM WRFROM
3155  013534  002354                          WRFROM                  ;STARTING FROM WRFROM
3156  013536  000106                          70.                     ;70. WORDS
3157  013540  177400                          177400                  ;FILL WITH 177400
3158
3159
3160                                          ;WRITE DATA COMMAND IS LOADED
3161
3162  013542  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3163  013546  000000                          0                       ;CYLINDER 0
3164  013550     001                  .BYTE   1                       ;SECTOR 1
3165  013551     000                  .BYTE   0                       ;TRACK 0
3166  013552  177672                          -70.                    ;WORD COUNT = 70.
3167  013554  002354                          WRFROM                  ;BUS ADDRESS
3168                                                                  ;STARTING ADDRESS OF DATA
3169                                                                  ;BUFFER = WRFROM
3170  013556  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3171  013560  010000                          FMT22                   ;16 BITS PER WORD FORMAT
3172                                                                  ;DO NOT INHIBIT ECC CORRECTION
3173                                                                  ;DO NOT INHIBIT HEADER COMPARE
3174  013562  002330                          WRIDAT                  ;GET READY TO DO A WRIDAT
3175                                                  ;WRITE DATA WITH 60 IN RHCS1
3176
3177                                          ;SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
3178  013564  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3179  013570  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3180  013572  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
3181                                                                  ;THE REGISTERS ARE SAVED
3182  013574  000022                          18.                     ;NUMBER OF REGISTERS
3183                                                                  ;SAVED = 18.
3184
3185  013576  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
3186
3187
3188  013602  013777  004472  166350          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3189                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3190                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3191                                                                  ;'TIME' WILL ONLY SAVE
3192                                                                  ;CURRENT CYLINDER ADDRESS
3193                                                                  ;AND LOOK AHEAD REGISTERS
```

```
3194
3195
3196  013610  013746  002330                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
3197  013614  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3198                                                                 ;ENABLE INTERRUPT
3199  013620  012677  166346                MOV     (SP)+,@RHCS1    ;GO WITH
3200                                                                  ;60 IN RHCS1 FOR WRITE DATA
3201                                                                 ;WITH INTERRUPT ENABLED
3202
3203                                        ;TIME IS NOT CRITICAL
3204
3205  013624  104420                        WAT                     ;WAIT FOR RDY BIT TO SET
3206  013626  002172                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
3207  013630  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3208  013632  001614                        900.                    ;ALLOW 9000 MICRO SECONDS
3209  013634  001502                        834.                    ;RDY MUST SET BETWEEN
3210                                                                 ;740 AND 17420 MICRO SECONDS
3211
3212                                        ;NOW CHANGE REGISTERS TO EXPECTED VALUE
3213  013636  004037  027762                JSR     R0,@#FILLRE     ;MOV -62. INTO SAVED RHWC
3214  013642  002164                        RHWC                    ;SAVED REGISTER TO CHANGE
3215  013644  177702                        -62.                    ;DATA
3216  013646  004037  027762                JSR     R0,@#FILLRE     ;MOV WRFROM+<8.*2> INTO SAVED RHBA
3217  013652  002166                        RHBA                    ;SAVED REGISTER TO CHANGE
3218  013654  002374                        WRFROM+<8.*2>                   ;DATA
3219
3220  013656  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3221  013662  002170                        RHCS2                   ;CHANGE RHCS2 REGISTER
3222
3223  013664  000002                        2                       ;2 BIT/BITS TO BE CHANGED
3224  013666  000001                        1                       ;NEW VALUE OF OR IS 1
3225  013670  000200                        OR                      ;CHANGE OR BIT
3226  013672  000000                        0                       ;NEW VALUE OF IR IS 0
3227  013674  000100                        IR                      ;CHANGE IR BIT
3228
3229  013676  004037  030732                JSP     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3230  013702  002172                        RHCS1                   ;CHANGE RHCS1 REGISTER
3231
3232  013704  000002                        2                       ;2 BIT/BITS TO BE CHANGED
3233  013706  000001                        1                       ;NEW VALUE OF SC IS 1
3234  013710  100000                        SC                      ;CHANGE SC BIT
3235  013712  000001                        1                       ;NEW VALUE OF TRE IS 1
3236  013714  040000                        TRE                     ;CHANGE TRE BIT
3237  013716  004037  027762                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
3238  013722  002176                        RHDST                   ;SAVED REGISTER TO CHANGE
3239  013724  000002                        2                       ;DATA
3240
3241  013726  053737  004622  004522        BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
3242                                                                 ;FOR WORKING DRIVE IN
3243                                                                 ;SAVED RHAS LOACTION
3244
3245  013734  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3246  013740  002174                        RHER1                   ;CHANGE RHER1 REGISTER
3247
3248  013742  000001                        1                       ;1 BIT/BITS TO BE CHANGED
3249  013744  000001                        1                       ;NEW VALUE OF HCE IS 1
```

```
3250  013746  000200                          HCE                     ;CHANGE HCE BIT
3251
3252  013750  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3253  013754  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
3254
3255  013756  000002                          2                       ;2 BIT/BITS TO BE CHANGED
3256  013760  000001                          1                       ;NEW VALUE OF ATA IS 1
3257  013762  100000                          ATA                     ;CHANGE ATA BIT
3258  013764  000001                          1                       ;NEW VALUE OF ERR IS 1
3259  013766  040000                          ERR                     ;CHANGE ERR BIT
3260
3261                                          ;COMPARE ALL REGISTERS
3262
3263  013770  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3264                                                                  ;PRESENT VALUE
3265  013774  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3266  013776  002242                          WC                      ;TEST DATA STARTING FROM 'PHWC'
3267  014000  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3268  014002  014006                          3$                      ;RETURN TO 3$ ON ERROR
3269  014004  014012                          4$                      ;RETURN TO 4$ ON NO ERROR
3270  014006  104052                  3$:     ERROR   52              ;WRITE DATA ON NON EXISTANT SECTOR
3271  014010  000207                                  RTS     PC      ;CAUSED IMPROPER REGISTER CHANGE
3272                                                                  ;ATTEMPTED WRITE WAS ON
3273                                                                  ;CYLINDER 0,SECTOR 1, TRACK 0
3274                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
3275                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
3276                                                                  ;AFTER COMMAND
3277
3278
3279                                          ;READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
3280                                          ;WILL BE ATTEMPTED
3281  014012                          4$:
3282
3283  014012  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3284                                                                  ;R3-RHDS1, R4-RHER1
3285                                                                  ;GIVE RH-11 INITIALIZE
3286                                                                  ;SETUP UNIT NUBER
3287
3288                                          ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
3289
3290  014016  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
3291  014022  002354                          WRFROM                  ;LOCATION WHERE SAVED
3292  014024  000006                          6                       ;NUMBER OF WORDS SAVED
3293  014026  010000                          10000                   ;FIRST DATA WORD
3294  014030  000000                          0                       ;SECOND DATA WORD
3295  014032  000000                          0                       ;THIRD DATA WORD
3296  014034  000000                          0                       ;FOURTH DATA WORD
3297  014036  000001                          1                       ;FIFTH DATA WORD
3298  014040  000001                          1                       ;SIXTH DATA WORD
3299  014042  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 198. WORDS, FROM WRFROM+<6*2>
3300  014046  002370                          WRFROM+<6*2>                    ;STARTING FROM WRFROM+<6*2>
3301  014050  000306                          198.                    ;198. WORDS
3302  014052  000000                          0                       ;FILL WITH 0
3303
3304  014054  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 50. WORDS, FROM WRFROM+<204.*2>
3305  014060  003204                          WRFROM+<204.*2>                 ;STARTING FROM WRFROM+<204.*2>
```

```
3306  014062  000062                          50.                     ;50. WORDS
3307  014064  177777                          -1                      ;FILL WITH -1
3308
3309                                          ;FILL READ INTO BUFFER WITH ALL ONES
3310  014066  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
3311  014072  003420                          REINTO                  ;STARTING FROM REINTO
3312  014074  000400                          256.                    ;256. WORDS
3313  014076  177777                          -1                      ;FILL WITH -1
3314
3315                                          ;FILL REGISTERS WITH READ HEADER AND DATA COMMAND
3316
3317  014100  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3318  014104  000000                          0                       ;CYLINDER 0
3319  014106     001                  .BYTE   1                       ;SECTOR 1
3320  014107     000                  .BYTE   0                       ;TRACK 0
3321  014110  177464                          -200.-4         ;WORD COUNT (DATA)=200.+
3322                                                                  ;4 HEADER WORDS
3323  014112  003420                          REINTO                  ;BUS ADDRESS
3324                                                                  ;STARTING ADDRESS OF DATA
3325                                                                  ;BUFFER = REINTO
3326  014114  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3327  014116  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
3328                                                                  ;INHIBIT ECC CORRECTION
3329                                                                  ;DO NOT INHIBIT HEADER COMPARE
3330  014120  002336                          REFOR                   ;GET READY TO DO A REFOR
3331                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
3332
3333
3334                                  ;SAVE REGISTERS FOR COMPARISON AFTER READ
3335                                  ;HEADER AND DATA
3336  014122  004037  030220          JSR     R0,@#SAVER      ;SAVE REGISTERS
3337  014126  002164                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3338  014130  004476                  SAVERE                  ;STARTING ADDRES OF WHERE
3339                                                          ;THE REGISTERS ARE SAVED
3340  014132  000022                  10.                     ;NUMBER OF REGISTERS
3341                                                          ;SAVED = 10.
3342
3343  014134  004737  030140          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
3344
3345  014140  013777  004472  166012  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3346                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
3347                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3348                                                          ;'TIME' WILL ONLY SAVE
3349                                                          ;CURRENT CYLINDER ADDRESS
3350                                                          ;AND LOOK AHEAD REGISTERS
3351
3352  014146  013746  002336          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
3353  014152  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3354                                                          ;ENABLE INTERRUPT
3355  014156  012677  166010          MOV     (SP)+,@RHCS1    ;GO WITH
3356                                                            ;72 IN RHCS1 FOR READ DATA
3357                                                          ;WITH INTERRUPT ENABLED
3358                                  ;TIME IS NOT CRITICAL
3359
3360  014162  104420                  WAT                     ;WAIT FOR RDY BIT TO SET
3361  014164  002172                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
3362  014166  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3363  014170  001614                          900.                    ;ALLOW 9000 MICRO SECONDS
3364  014172  001507                          839.                    ;RDY MUST SET BETWEEN
3365                                                                  ;690 AND 17470 MICRO SECONDS
3366                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
3367  014174  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3368  014200  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
3369  014202  000000                          0                       ;DATA
3370  014204  004037  027762                  JSR     R0,@#FILLRE     ;MOV REINTO+<204.*2> INTO SAVED RHBA
3371  014210  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
3372  014212  004250                          REINTO+<204.*2>                 ;DATA
3373
3374  014214  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3375  014220  002174                          RHER1                   ;CHANGE RHER1 REGISTER
3376
3377  014222  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3378  014224  000001                          1                       ;NEW VALUE OF HCE IS 1
3379  014226  000200                          HCE                     ;CHANGE HCE BIT
3380
3381  014230  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3382  014234  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
3383
3384  014236  000002                          2                       ;2 BIT/BITS TO BE CHANGED
3385  014240  000001                          1                       ;NEW VALUE OF ATA IS 1
3386  014242  100000                          ATA                     ;CHANGE ATA BIT
3387  014244  000001                          1                       ;NEW VALUE OF ERR IS 1
3388  014246  040000                          ERR                     ;CHANGE ERR BIT
3389  014250  004037  027762                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
3390  014254  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
3391  014256  000002                          2                       ;DATA
3392
3393  014260  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3394  014264  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
3395
3396  014266  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3397  014270  000001                          1                       ;NEW VALUE OF SC!TRE IS 1
3398  014272  140000                          SC!TRE                  ;CHANGE SC!TRE BIT
3399
3400  014274  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
3401                                                                  ;FOR WORKING DRIVE IN
3402                                                                  ;SAVED RHAS LOACTION
3403
3404                                          ;COMPARE REGISTERS BEFORE READ HEADER AND DATA
3405                                          ;WITH AFTER
3406
3407  014302  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3408                                                                  ;PRESENT VALUE
3409  014306  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3410  014310  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3411  014312  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3412  014314  014320                          5$                      ;RETURN TO 5$ ON ERROR
3413  014316  014324                          6$                      ;RETURN TO 6$ ON NO ERROR
3414
3415  014320  104031                  5$:     ERROR   31              ;READ HEADER AND DATA WITH
3416  014322  000207                          RTS     PC              ;FORCED HEADER COMPARE ERROR
3417                                                                  ;CAUSED ERROR
```

```
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER READ

                                        ;NOW COMPARE READ DATA
                                        ;THE COMMAND READ ONLY 204 WORDS, 4 HEADER WORDS
                                        ;AND 200 DATA WORDS
014324                          6$:

014324  004037  032070                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
014330  002354                        WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
014332  003420                        REINTO                  ;TEST DATA STARTS FROM REINTO
014334  000400                        256.                    ;256. WORDS TO BE COMPARED
014336  014342                        7$                      ;RETURN TO 7$ ON ERROR
014340  014346                        8$                      ;RETURN TO 8$ ON NO ERROR

014342  104034                7$:     ERROR   34              ;DATA READ FROM A FORCED
014344  000207                        RTS     PC              ;HEADER COMPARE ERROP IS
                                                        ;INCORRECT
                                                        ;GOOD DATA GIVES WHAT
                                                        ;THE READ HEADER AND DATA
                                                        ;SHOULD HAVE READ
                                                        ;BAD DATA GIVES WHAT
                                                        ;WAS IN BUFFER AFTER
                                                        ;READ COMMAND
014346                          8$:


                                ;*********************************************************************
                                ;*TEST 15       HEADER COMPARE ERROR - RHER1 BIT #7

                                ;*      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

                                ;*********************************************************************
014346  000004                  TST15:  SCOPE
014350  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014354  012737  000015  004470          MOV     #15,@#TSTNM     ;SAVE TEST NUMBER

014362  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUBER

                                        ;GET HEADS TO CYLINDER 0
014366  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
```

```
3474  014372  013777  004472  165560          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3475                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3476                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3477                                                                  ;'TIME' WILL ONLY SAVE
3478                                                                  ;CURRENT CYLINDER ADDRESS
3479                                                                  ;AND LOOK AHEAD REGISTERS
3480
3481  014400  013746  002314                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
3482  014404  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3483                                                                  ;ENABLE INTERRUPT
3484  014410  012677  165556                  MOV     (SP)+,@RHCS1    ;GO WITH
3485                                                                   ;6 IN RHCS1 FOR RECALIBRATE
3486                                                                  ;WITH INTERRUPT ENABLED
3487  014414  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3488  014416  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3489
3490  014420  104420                          WAT                     ;WAIT FOR DRY BIT TO SET
3491  014422  002214                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
3492  014424  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3493  014426  012740                          5600.                   ;ALLOW 56000 MICRO SECONDS
3494  014430  012737                          5599.                   ;DRY MUST SET BETWEEN
3495                                                                  ;10 AND 111990 MICRO SECONDS
3496
3497  014432  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3498                                                                  ;R3-RHDS1, R4-RHER1
3499                                                                  ;GIVE RH-11 INITIALIZE
3500                                                                  ;SETUP UNIT NUBER
3501                                          ;FILL REGISTERS FOR SEARCH
3502
3503  014436  004037  030010                  JSR     R0,@#SRCH       ;SEARCH FOR
3504  014442  000000                          0                       ;CYLINDER 0
3505  014444     001                  .BYTE   1                       ;SECTOR 1
3506  014445     000                  .BYTE   0                       ;TRACK 0
3507
3508
3509                                          ;SAVE REGISTERS FOR COMPARISON AFTER SEARCH
3510  014446  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3511  014452  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3512  014454  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
3513                                                                  ;THE REGISTERS ARE SAVED
3514  014456  000022                          18.                     ;NUMBER OF REGISTERS
3515                                                                  ;SAVED = 18.
3516
3517
3518  014460  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
3519
3520
3521  014464  013777  004472  165466          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3522                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3523                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3524                                                                  ;'TIME' WILL ONLY SAVE
3525                                                                  ;CURRENT CYLINDER ADDRESS
3526                                                                  ;AND LOOK AHEAD REGISTERS
3527
3528
3529  014472  013746  002322                  MOV     @#SERCH,-(SP)   ;GET READY TO MOVE COMMAND
```

```
3530  014476  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3531                                                                    ;ENABLE INTERRUPT
3532  014502  012677  165464                  MOV     (SP)+,@RHCS1    ;GO WITH
3533                                                                    ;WITH INTERRUPT ENABLED
3534  014506  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3535  014510  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3536
3537
3538  014512  104420                          WAT                     ;WAIT FOR DPY BIT TO SET
3539  014514  002214                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
3540  014516  000200                          DRY                     ;WAIT FOR DPY BIT IN RHDS1 REGISTER
3541  014520  001614                          900.                    ;ALLOW 9000 MICRO SECONDS
3542  014522  001507                          839.                    ;DRY MUST SET BETWEEN
3543                                                                    ;690 AND 17470 MICRO SECONDS
3544                                          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
3545                                          ;R0 AND R5 IMMEDIATELY AFTER GO
3546  014524  013746  002322                  MOV     @#SERCH,-(SP)   ;SAVE COMMAND
3547  014530  052716  004301                  BIS     #IE!GO!DVA!RDY,(SP)     ;INCLUDE IE!GO!DVA!RDY
3548  014534  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3549  014540  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!RDY
3550                                                                  ;AND COMMAND SHOULD BE SET
3551  014542  001405                          BEQ     67$             ;BRANCH IF GOOD
3552  014544  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
3553  014550  010137  004464                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3554  014554  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
3555                                                                  ;COMMAND AND IE!GO!DVA!RDY SHOULD BE SET
3556  014556  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
3557  014562  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3558  014566  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
3559                                                                  ;SHOULD BE SET
3560  014570  001405                          BEQ     69$             ;BRANCH IF GOOD
3561  014572  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
3562  014576  010337  004464                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
3563  014602  104063                          ERROP   63              ;DURING ABOVE OPERATION ONLY
3564                                                                  ;MOL!DPR!VV SHOULD BE SET
3565  014604                          69$:
3566
3567                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
3568
3569  014604  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3570  014610  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
3571
3572  014612  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3573  014614  000001                          1                       ;NEW VALUE OF SC IS 1
3574  014616  100000                          SC                      ;CHANGE SC BIT
3575
3576  014620  004037  030732                  JSR     R2,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3577  014624  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
3578
3579  014626  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3580  014630  000001                          1                       ;NEW VALUE OF ATA IS 1
3581  014632  100000                          ATA                     ;CHANGE ATA BIT
3582
3583  014634  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
3584                                                                  ;FOR WORKING DRIVE IN
3585                                                                  ;SAVED RHAS LOACTION
```

```
                                        ;COMPARE REGISTERS BEFORE SEARCH WITH AFTER SEARCH
014642  004037  031040                  JSR     P0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
014646  004476                          SAVERF                  ;GOOD DATA SAVED IN 'SAVERF'
014650  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
014652  000022                          18.                     ;18. REGISTERS TO BE COMPARED
014654  014660                          1$                      ;RETURN TO 1$ ON ERROR
014656  014664                          2$                      ;RETURN TO 2$ ON NO ERROR

014660  104047                  1$:     ERROR   47              ;SEARCH TO A NON-EXISTANT

014662  000207                          RTS     PC              ;SECTOR CAUSED IMPROPER
                                                                ;REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES
                                                                ;WHAT WAS THERE AFTER
                                                                ;SEARCH
014664                          2$:

                                ;*****************************************************************
                                ;*TEST 16       RESTORE SECTOR 1 CYLINDER 1 TRACK 1

                                ;*      THIS REPLACES REMOVER 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.

                                ;*****************************************************************
014664  000004                  TST16:  SCOPE
014666  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014672  012737  000016  004470          MOV     #16,@#TSTNM     ;SAVE TEST NUMBER
```

```
3642  014700  004737  030060                  JSR     PC,@#CLDISK     ;SET R1=RHCS1, R2=RHCS2
3643                                                                  ;R3=RHDS1, R4=RHER1
3644                                                                  ;GIVE RH-11 INITIALIZE
3645                                                                  ;SETUP UNIT NUBER
3646
3647                                          ;FILL WRITE FROM BUFFER WITH HEADER
3648
3649  014704  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
3650  014710  002354                          WRFROM                  ;LOCATION WHERE SAVED
3651  014712  000004                          4                       ;NUMBER OF WORDS SAVED
3652  014714  010000                          10000                   ;FIRST DATA WORD
3653  014716  000001                          1                       ;SECOND DATA WORD
3654  014720  000000                          0                       ;THIRD DATA WORD
3655  014722  000000                          0                       ;FOURTH DATA WORD
3656
3657                                          ;FILL WRITE FROM BUFFER WITH DATA
3658  014724  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
3659  014730  002364                          WRFROM+10                       ;STARTING FROM WRFROM+10
3660  014732  000400                          256.                    ;256. WORDS
3661  014734  000000                          0                       ;FILL WITH 0
3662
3663
3664                                          ;NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
3665                                          ;AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
3666                                          ;CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
3667                                          ;CHANGE WRITE FROM BUFFER
3668
3669  014736  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
3670  014742  003420                          REINTO                  ;LOCATION WHERE SAVED
3671  014744  000004                          4                       ;NUMBER OF WORDS SAVED
3672  014746  010000                          10000                   ;FIRST DATA WORD
3673  014750  000001                          1                       ;SECOND DATA WORD
3674  014752  000000                          0                       ;THIRD DATA WORD
3675  014754  000000                          0                       ;FOURTH DATA WORD
3676  014756  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
3677  014762  003430                          REINTO+10                       ;STARTING FROM REINTO+10
3678  014764  000400                          256.                    ;256. WORDS
3679  014766  000000                          0                       ;FILL WITH 0
3680
3681
3682                                          ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
3683
3684  014770  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3685  014774  000000                          0                       ;CYLINDER 0
3686  014776     001                  .BYTE   1                       ;SECTOR 1
3687  014777     000                  .BYTE   0                       ;TRACK 0
3688  015000  177374                          -256.-4         ;WORD COUNT (DATA)=256.+
3689                                                                  ;4 HEADER WORDS
3690  015002  002354                          WRFROM                  ;BUS ADDRESS
3691                                                                  ;STARTING ADDRESS OF DATA
3692                                                                  ;BUFFER = WRFROM
3693  015004  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3694  015006  010000                          FMT22                   ;16 BITS PER WORD FORMAT
3695                                                                  ;DO NOT INHIBIT ECC CORRECTION
3696                                                                  ;DO NOT INHIBIT HEADER COMPARE
3697  015010  002332                          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
015012  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
015016  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
015020  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
015022  000021                          17.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 17.

015024  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV

015030  013777  004472  165122          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS

015036  013746  002332                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
015042  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
015046  012677  165120                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED
015052  011100                  MOV     @R1,R0                  ;SAVE RHCS1 DURING ABOVE OPERATION
015054  011305                  MOV     @R3,R5                  ;SAVE RHDS1 DURING ABOVE OPERATION
                                ;ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC

015056  104420                  WAT                             ;WAIT FOR RDY BIT TO SET
015060  002172                  RHCS1                           ;WAIT FOR RHCS1 REGISTER
015062  000200                  RDY                             ;WAIT FOR RDY BIT IN RHCS1 REGISTER
015064  001614                  900.                            ;ALLOW 9000 MICRO SECONDS
015066  001507                  839.                            ;RDY MUST SET BETWEEN
                                                                ;690 AND 17470 MICRO SECONDS

                                ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                ;R0 AND R5 IMMEDIATELY AFTER GO
015070  013746  002332                  MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
015074  052716  000101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
015100  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
015104  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
015106  001405                          BEQ     64$             ;BRANCH IF GOOD
015110  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
015114  010137  000464                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
015120  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
015122  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
015126  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
015132  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
015134  001405                          BEQ     66$             ;BRANCH IF GOOD
015136  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
```

```
3754  015142  010337  004464                  MOV     P3,@#REGADR     ;FAILING REGISTER RHDS1
3755  015146  104063                          ERROP   63              ;DURING ABOVE OPERATION ONLY
3756                                                                  ;MOL!DPR!VV SHOULD BE SET
3757  015150                          66$:
3758
3759                                          ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
3760  015150  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3761  015154  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
3762  015156  000000                          0                       ;DATA
3763  015160  004037  027762                  JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2> INTO SAVED RHBA
3764  015164  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
3765  015166  003364                          WRFROM+<260.*2>                 ;DATA
3766  015170  004037  027762                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
3767  015174  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
3768  015176  000002                          2                       ;DATA
3769                                          ;NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
3770                                          ;WITH REGISTERS AFTER COMMAND
3771
3772  015200  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3773                                                                  ;PRESENT VALUE
3774  015204  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3775  015206  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3776  015210  000021                          17.                     ;17. REGISTERS TO BE COMPARED
3777  015212  015216                          18                      ;RETURN TO 18 ON ERROR
3778  015214  015222                          28                      ;RETURN TO 28 ON NO ERROR
3779
3780  015216  104027                  18:     ERROR   27              ;WRITE HEADER AND DATA
3781  015220  000207                          RTS     PC              ;CAUSED IMPROPER REGISTER
3782                                                                  ;CHANGE
3783                                                                  ;GOOD DATA GIVES WHAT SHOULD
3784                                                                  ;BE THERE
3785                                                                  ;RECEIVED DATA GIVES WHAT
3786                                                                  ;WAS THERE AFTER COMMAND
3787
3788                                          ;NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
3789                                          ;NOTHER GOT CHANGED
3790  015222                          28:
3791
3792  015222  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
3793  015226  003420                          REINTO                  ;GOOD DATA      STARTS FROM REINTO
3794  015230  002354                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
3795  015232  000404                          260.                    ;260. WORDS TO BE COMPARED
3796  015234  015240                          38                      ;RETURN TO 38 ON ERROR
3797  015236  015244                          48                      ;RETURN TO 48 ON NO ERROR
3798
3799
3800  015240  104030                  38:     ERROR   30              ;WRITE HEADER AND DATA
3801  015242  000207                          RTS     PC              ;CHANGED WRITE FROM BUFFER
3802
3803                                          ;NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
3804                                          ;READ INTO BUFFER IS FILLED WITH ONES
3805  015244                          48:
3806
3807  015244  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3808                                                                  ;R3-RHDS1, R4-RHER1
3809                                                                  ;GIVE RH-11 INITIALIZE
```

```
3810                                                    ;SETUP UNIT NUBER
3811  015250  004037  027730                JSR     P0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
3812  015254  003420                        REINTO                  ;STARTING FROM REINTO
3813  015256  000404                        260.                    ;260. WORDS
3814  015260  177777                        -1                      ;FILL WITH -1
3815
3816
3817                                        ;NOW FILL COMMAND
3818
3819  015262  004037  032024                JSP     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3820  015266  000000                        0                       ;CYLINDER 0
3821  015270     001                .BYTE   1                       ;SECTOR 1
3822  015271     000                .BYTE   0                       ;TRACK 0
3823  015272  177374                        -256.-4         ;WORD COUNT (DATA)=256.+
3824                                                          ;4 HEADER WORDS
3825  015274  003420                        REINTO                  ;BUS ADDRESS
3826                                                          ;STARTING ADDRESS OF DATA
3827                                                          ;BUFFER = REINTO
3828  015276  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3829  015300  014000                        ECI!FMT22                       ;16 BITS PER WORD FORMAT
3830                                                          ;INHIBIT ECC CORRECTION
3831                                                          ;DO NOT INHIBIT HEADER COMPARE
3832  015302  002336                        REFOP                   ;GET READY TO DO A REFOR
3833                                                          ;READ HEADER AND DATA WITH 72 IN RHCS1
3834
3835
3836                                        ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
3837  015304  004037  030220                JSR     R0,@#SAVER      ;SAVE REGISTERS
3838  015310  002164                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3839  015312  004476                        SAVEPE                  ;STARTING ADDRES OF WHERE
3840                                                          ;THE REGISTERS ARE SAVED
3841  015314  000022                        18.                     ;NUMBER OF REGISTERS
3842                                                          ;SAVED = 18.
3843
3844  015316  004737  030140                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
3845
3846
3847  015322  013777  004472  164630        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3848                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
3849                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3850                                                          ;'TIME' WILL ONLY SAVE
3851                                                          ;CURRENT CYLINDER ADDRESS
3852                                                          ;AND LOOK AHEAD REGISTERS
3853
3854
3855  015330  013746  002336                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
3856  015334  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3857                                                          ;ENABLE INTERRUPT
3858  015340  012677  164626                MOV     (SP)+,@RHCS1    ;GO WITH
3859                                                            ;72 IN RHCS1 FOR READ DATA
3860                                                          ;WITH INTERRUPT ENABLED
3861  015344  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3862  015346  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3863
3864
3865  015350  104420                        WAT                     ;WAIT FOR RDY BIT TO SET
```

```
3866  015352  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
3867  015354  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3868  015356  001614                          900.                    ;ALLOW 9000 MICRO SECONDS
3869  015360  001507                          839.                    ;RDY MUST SET BETWEEN
3870                                                                  ;690 AND 17470 MICRO SECONDS
3871                                          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
3872                                          ;R0 AND R5 IMMEDIATELY AFTER GO
3873  015362  013746  002336                  MOV     @#REFOR,-(SP)   ;SAVE COMMAND
3874  015366  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
3875  015372  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3876  015376  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
3877                                                                  ;AND COMMAND SHOULD BE SET
3878  015400  001405                          BEQ     67$             ;BRANCH IF GOOD
3879  015402  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
3880  015406  010137  004464                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3881  015412  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
3882                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
3883  015414  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
3884  015420  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3885  015424  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
3886                                                                  ;SHOULD BE SET
3887  015426  001405                          BEQ     69$             ;BRANCH IF GOOD
3888  015430  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
3889  015434  010337  004464                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
3890  015440  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
3891                                                                  ;MOL!DPR!VV SHOULD BE SET
3892  015442                          69$:
3893
3894                                          ;CHEANGE SAVED REGISTERS TO EXPECTED VALUES
3895  015442  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3896  015446  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
3897  015450  000000                          0                       ;DATA
3898  015452  004037  027762                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
3899  015456  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
3900  015460  004430                          REINTO+<260.*2>                 ;DATA
3901  015462  004037  027762                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
3902  015466  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
3903  015470  000002                          2                       ;DATA
3904
3905                                          ;COMPARE REGISTER BEFORE READ HEADER AND DATA
3906                                          ;WITH REGISTERS AFTER COMMAND
3907
3908
3909  015472  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3910                                                                  ;PRESENT VALUE
3911  015476  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3912  015500  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3913  015502  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3914  015504  015510                          5$                      ;RETURN TO 5$ ON ERROR
3915  015506  015514                          6$                      ;RETURN TO 6$ ON NO ERROR
3916
3917  015510  104031                  5$:     ERROR   31              ;READ HEADER AND DATA CAUSED
3918  015512  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
3919                                                                  ;GOOD DATA GIVES WHAT SHOULD
3920                                                                  ;BE THERE
3921                                                                  ;RECEIVED DATA GIVES WHAT WAS
```

```
                                                        ;THERE AFTER COMMAND

                                ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                                ;THE READ WAS GOOD
015514                      6$:

015514  004037  032070            JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
015520  002354                    WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
015522  003420                    REINTO                  ;TEST DATA STARTS FROM REINTO
015524  000404                    260.                    ;260. WORDS TO BE COMPARED
015526  015532                    7$                      ;RETURN TO 7$ ON ERROR
015530  015536                    10$                     ;RETURN TO 10$ ON NO ERROR

015532  104032              7$:   ERROR   32              ;WRITE HEADER AND DATA
015534  000207                    RTS     PC              ;FOLLOWED BY A READ HEADER
                                                        ;AND DATA GAVE A READ ERROR
                                                        ;ERROR MAY BE IN READ OR WRITE
015536                      10$:


                        ;*****************************************************************
                        ;*TEST 17       INVALID ADDRESS ERROR RHER1-BIT #10

                        ;*      A WRITE HEADER AND DATA WILL BE ATTEMPTED TO CYLINDER 411
                        ;*      TRACK 0, SECTOR 0
                        ;*      INVALID ADDRESS ERROR (IAE) BIT #10 IN RHER1 SHOULD
                        ;*      SET

                        ;*****************************************************************
015536  000004        TST17:  SCOPE
015540  012706  001000            MOV     #STACK,SP       ;RESET STACK
015544  012737  000017  004470    MOV     #17,@#TSTNM     ;SAVE TEST NUMBER

015552  004737  030060            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUBER

                                ;FILL WRITE FROM BUFFER WITH HEADER

015556  004037  027704            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
015562  002354                    WRFROM                  ;LOCATION WHERE SAVED
015564  000004                    4                       ;NUMBER OF WORDS SAVED
015566  010633                    10633                   ;FIRST DATA WORD
015570  000000                    0                       ;SECOND DATA WORD
015572  000000                    0                       ;THIRD DATA WORD
015574  000000                    0                       ;FOURTH DATA WORD

                                ;FILL WRITE FROM BUFFER WITH DATA
015576  004037  027730            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
015602  002364                    WRFROM+10                       ;STARTING FROM WRFROM+10
015604  000400                    256.                    ;256. WORDS
015606  177777                    0-1                     ;FILL WITH 0-1
```

```
                                        ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
                                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
015614  000633                          633                     ;CYLINDER 633
015616     000                  .BYTE   0                       ;SECTOR 0
015617     000                  .BYTE   0                       ;TRACK 0
015620  177374                          -256.-4         ;WORD COUNT (DATA)=256.-
                                                                ;4 HEADER WORDS
015622  002354                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
015624  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
015626  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
015630  002332                          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
015632  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
015636  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
015640  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
015642  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

015644  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV

015650  013777  004472  164302          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS

015656  013746  002332                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
015662  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
015666  012677  164300                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED


015672  104420                          WAT                     ;WAIT FOR IAE BIT TO SET
015674  002174                          RHER1                   ;WAIT FOR RHER1 REGISTER
015676  002000                          IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
015700  000011                          9.                      ;ALLOW 90 MICRO SECONDS
015702  000011                          9.                      ;IAE MUST SET BETWEEN
                                                                ;00 AND 180 MICRO SECONDS

                                        ;CHANGE SAVE REGISTERS TO EXPECTED VALUE
```

```
4034
4035
4036
4037                                    ;AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4038                                    ;RHWC,RHBA,RHCS1,RHCS2,  CANNOT BE PEREDETERMINED
4039                                    ;THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4040  015704  017737  164254  004476    MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
4041                                                            ;AS EXPLAINED ABOVE
4042  015712  017737  164250  004500    MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
4043                                                            ;AS EXPLAINED ABOVE
4044  015720  017737  164244  004502    MOV     @RHCS2,@#SAVERE+4;RHCS2 IS UNPREDICTABLE
4045                                                            ;AS EXPLAINED ABOVE
4046  015726  017737  164240  004504    MOV     @RHCS1,@#SAVERE+6;RHCS1 IS UNPREDICTABLE
4047                                                            ;AS EXPLAINED ABOVE
4048
4049  015734  004037  030732            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4050  015740  002214                    RHDS1                   ;CHANGE RHDS1 REGISTER
4051
4052  015742  000002                    2                       ;2 BIT/BITS TO BE CHANGED
4053  015744  000001                    1                       ;NEW VALUE OF ATA IS 1
4054  015746  100000                    ATA                     ;CHANGE ATA BIT
4055  015750  000001                    1                       ;NEW VALUE OF ERR IS 1
4056  015752  040000                    ERR                     ;CHANGE ERR BIT
4057
4058  015754  004037  030732            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4059  015760  002174                    RHER1                   ;CHANGE RHER1 REGISTER
4060
4061  015762  000001                    1                       ;1 BIT/BITS TO BE CHANGED
4062  015764  000001                    1                       ;NEW VALUE OF IAE IS 1
4063  015766  002000                    IAE                     ;CHANGE IAE BIT
4064
4065  015770  053737  004622  004522    BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
4066                                                            ;FOR WORKING DRIVE IN
4067                                                            ;SAVED RHAS LOACTION
4068  015776  017737  164174  004510    MOV     @RHDST,@#SAVERE+12      ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
4069
4070
4071                                    ;COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH
4072                                    ;AFTER ATTEMPTED WRITE WITH A IAE ERROR
4073
4074  016004  004037  031040            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4075                                                            ;PRESENT VALUE
4076  016010  004476                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4077  016012  002242                    WC                      ;TEST DATA STARTING FROM 'RHWC'
4078  016014  000022                    18.                     ;18. REGISTERS TO BE COMPARED
4079  016016  016022                    1$                      ;RETURN TO 1$ ON ERROR
4080  016020  016026                    2$                      ;RETURN TO 2$ ON NO ERROR
4081
4082  016022  104054            1$:     ERROR   54              ;ATTEMPTED OPERATION WITH
4083  016024  000207                    RTS     PC              ;INVALID ADDRESS CAUSED
4084                                                            ;IMPROPER REGISTER CHANGE
4085                                                            ;GOOD DATA GIVES WHAT SHOULD
4086                                                            ;BE THERE
4087  016026                    2$:                             ;RECEIVED DATA GIVES REGISTER
4088                                                            ;CONTENTS AFTER ATTEMPTED
4089                                                            ;WRITE HEADER AND DATA
```

```
                                ;**********************************************************************
                                ;*TEST 20       INVALID ADDRESS ERROR RHER1 - BIT #10
                                ;
                                ;*      A WRITE DATA IS ATTEMPTED TO CYLINDER 0, TRACK 19,
                                ;*      SECTOR 0
                                ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
                                ;*      SHOULD SET
                                ;
                                ;**********************************************************************
016026  000004                  TST20:  SCOPE
016030  012706  001000                  MOV     #STACK,SP       ;RESET STACK
016034  012737  000020  004470          MOV     #20,@#TSTNM     ;SAVE TEST NUMBER
                                ;
016042  004737  030060                  JSR     PC,@#CLDISK     ;SET P1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                ;
                                        ;FILL WRITE FROM BUFFER WITH DATA
016046  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
016052  002354                  WRFROM                          ;STARTING FROM WRFROM
016054  000400                  256.                            ;256. WORDS
016056  000377                  377                             ;FILL WITH 377
                                ;
                                        ;WRITE DATA COMMAND WILL BE FILLED
                                ;
016060  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
016064  000000                  0                               ;CYLINDER 0
016066     000          .BYTE   0                               ;SECTOR 0
016067     023          .BYTE   19.                             ;TRACK 19.
016070  177400                  -256.                           ;WORD COUNT = 256.
016072  002354                  WRFROM                          ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
016074  000000                  0                               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
016076  010000                  FMT22                           ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
016100  002330                  WRIDAT                          ;GET READY TO DO A WRIDAT
                                                        ;WRITE DATA WITH 60 IN RHCS1
                                ;
                                ;
                                        ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
016102  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
016106  002164                  RHWC                            ;RHWC IS THE FIRST REGISTER SAVED
016110  004476                  SAVERE                          ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
016112  000022                  18.                             ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.
                                ;
016114  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
                                ;
                                ;
```

```
4146  016120  013777  004472  164032          MOV     @#RP4VEC,@RPVEC ;SET PPR4 VECTOR ADDRESS
4147                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
4148                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4149                                                                  ;'TIME' WILL ONLY SAVE
4150                                                                  ;CURRENT CYLINDER ADDRESS
4151                                                                  ;AND LOOK AHEAD REGISTERS
4152
4153
4154  016126  013746  002330                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
4155  016132  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4156                                                                  ;ENABLE INTERRUPT
4157  016136  012677  164030                  MOV     (SP)+,@RHCS1    ;GO WITH
4158                                                                   ;60 IN RHCS1 FOR WRITE DATA
4159                                                                  ;WITH INTERRUPT ENABLED
4160
4161
4162  016142  104420                          WAT                     ;WAIT FOR IAE BIT TO SET
4163  016144  002174                          RHER1                   ;WAIT FOR RHER1 REGISTER
4164  016146  002000                          IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
4165  016150  000011                          9.                      ;ALLOW 90 MICRO SECONDS
4166  016152  000011                          9.                      ;IAE MUST SET BETWEEN
4167                                                                  ;90 AND 180 MICRO SECONDS
4168
4169                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4170
4171
4172
4173                                          ;AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4174                                          ;RHWC,RHBA,RHCS1,RHCS2,  CANNOT BE PEREDETERMINED
4175                                          ;THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4176  016154  017737  164004  004476          MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
4177                                                                  ;AS EXPLAINED ABOVE
4178  016162  017737  164000  004500          MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
4179                                                                  ;AS EXPLAINED ABOVE
4180  016170  017737  163774  004502          MOV     @RHCS2,@#SAVERE+4;RHCS2 IS UNPREDICTABLE
4181                                                                  ;AS EXPLAINED ABOVE
4182  016176  017737  163770  004504          MOV     @RHCS1,@#SAVERE+6;RHCS1 IS UNPREDICTABLE
4183                                                                  ;AS EXPLAINED ABOVE
4184
4185  016204  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4186  016210  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
4187
4188  016212  000002                          2                       ;2 BIT/BITS TO BE CHANGED
4189  016214  000001                          1                       ;NEW VALUE OF ATA IS 1
4190  016216  100000                          ATA                     ;CHANGE ATA BIT
4191  016220  000001                          1                       ;NEW VALUE OF ERR IS 1
4192  016222  040000                          ERR                     ;CHANGE ERR BIT
4193  016224  017737  163746  004510          MOV     @RHDST,@#SAVERE+12      ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
4194
4195
4196  016232  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4197  016236  002174                          RHER1                   ;CHANGE RHER1 REGISTER
4198
4199  016240  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4200  016242  000001                          1                       ;NEW VALUE OF IAE IS 1
4201  016244  002000                          IAE                     ;CHANGE IAE BIT
```

```
4202
4203  016246  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
4204                                                                            ;FOR WORKING DRIVE IN
4205                                                                            ;SAVED RHAS LOACTION
4206
4207                                          ;COMPARE REGISTERS BEFORE ATTEMPTED WRITE DATA
4208                                          ;WITH AFTER ATTEMPT, IAE SHOULD BE SET
4209
4210  016254  004037  031040                 JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4211                                                                  ;PRESENT VALUE
4212  016260  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4213  016262  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4214  016264  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4215  016266  016272                          18                      ;RETURN TO 18 ON ERROR
4216  016270  016276                          28                      ;RETURN TO 28 ON NO ERROR
4217
4218  016272  104054                  18:     ERROR   54              ;ATTEMPTED WRITE DATA
4219  016274  000207                          RTS     PC              ;WITH INVALID ADDRESS
4220                                                                  ;CAUSED IMPPOPER REGISTER
4221                                                                  ;CHANGE
4222  016276                          28:                             ;GOOD DATA GIVES WHAT
4223                                                                  ;SHOULD BE THERE
4224                                                                  ;RECEIVED DATA GIVES WHAT
4225                                                                  ;WAS THERE AFTER AFTER ATTEMPT
4226
4227
4228                                  ;*****************************************************************
4229                                  ;*TEST 21       INVALID ADDRESS ERROR RHER1 -BIT #10
4230
4231                                  ;*      A READ HEADER AND DATA IS ATTEMPTED TO CYLINDER 0
4232                                  ;*      TRACK 0, SECTOR 22
4233                                  ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
4234                                  ;*      SHOULD SET
4235                                  ;*      THIS WILL START WITH THE HEADS ON CYLINDER 10
4236                                  ;*      TO PROVE THAT IAE SETS EVEN BEFORE THE IMPLIED
4237                                  ;*      SEEK
4238
4239                                  ;*****************************************************************
4240  016276  000004                  TST21:  SCOPE
4241  016300  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4242  016304  012737  000021  004470          MOV     #21,@#TSTNM     ;SAVE TEST NUMBER
4243
4244  016312  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4245                                                                  ;R3-RHDS1, R4-RHER1
4246                                                                  ;GIVE RH-11 INITIALIZE
4247                                                                  ;SETUP UNIT NUBER
4248
4249                                          ;GET THE HEADS TO CYLINDER 10
4250  016316  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
4251
4252  016322  013777  004472  163630          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4253                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
4254                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4255                                                                  ;'TIME' WILL ONLY SAVE
4256                                                                  ;CURRENT CYLINDER ADDRESS
4257                                                                  ;AND LOOK AHEAD REGISTERS
```

```
4258
4259  016330  004037  030030                  JSR     R0,@#SEEKCY     ;SEEK FOR
4260  016334  000010                          10                      ;CYLINDER 10
4261
4262  016336  013746  002340                  MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
4263  016342  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4264                                                                  ;ENABLE INTERRUPT
4265  016346  012677  163620                  MOV     (SP)+,@RHCS1    ;GO WITH
4266                                                                   ;4 IN RHCS1 FOR SEEK
4267                                                                  ;WITH INTERRUPT ENABLED
4268
4269  016352  104420                          WAT                     ;WAIT FOR DRY BIT TO SET
4270  016354  002214                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
4271  016356  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4272  016360  015530                          7000.                   ;ALLOW 70000 MICRO SECONDS
4273  016362  015530                          7000.                   ;DRY MUST SET BETWEEN
4274                                                                  ;00 AND 140000 MICRO SECONDS
4275
4276  016364  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4277                                                                  ;R3-RHDS1, R4-RHER1
4278                                                                  ;GIVE RH-11 INITIALIZE
4279                                                                  ;SETUP UNIT NUBER
4280                                          ;FILL READ INTO BUFFER WITH 125252
4281  016370  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 260 WORDS, FROM REINTO
4282  016374  003420                          REINTO                  ;STARTING FROM REINTO
4283  016376  000260                          260                     ;260 WORDS
4284  016400  125252                          125252                  ;FILL WITH 125252
4285
4286
4287                                          ;THE READ HEADER AND DATA COMMAND IS FILLED
4288
4289  016402  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
4290  016406  000000                          0                       ;CYLINDER 0
4291  016410     026                  .BYTE   22.                     ;SECTOR 22.
4292  016411     000                  .BYTE   0                       ;TRACK 0
4293  016412  177374                          -256.-4         ;WORD COUNT (DATA)=256.+
4294                                                                  ;4 HEADER WORDS
4295  016414  003420                          REINTO                  ;BUS ADDRESS
4296                                                                  ;STARTING ADDRESS OF DATA
4297                                                                  ;BUFFER = REINTO
4298  016416  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4299  016420  014000                          FMT22!ECI                       ;16 BITS PER WORD FORMAT
4300                                                                  ;INHIBIT ECC CORRECTION
4301                                                                  ;DO NOT INHIBIT HEADER COMPARE
4302  016422  002336                          REFOR                   ;GET READY TO DO A REFOR
4303                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
4304
4305
4306                                          ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
4307  016424  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
4308  016430  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4309  016432  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
4310                                                                  ;THE REGISTERS ARE SAVED
4311  016434  000022                          18.                     ;NUMBER OF REGISTERS
4312                                                                  ;SAVED = 18.
4313
```

```
4314  016436  004737  030140                  JSP     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
4315
4316
4317  016442  013777  004472  163510          MOV     @#PP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4318                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
4319                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4320                                                                  ;'TIME' WILL ONLY SAVE
4321                                                                  ;CURRENT CYLINDER ADDRESS
4322                                                                  ;AND LOOK AHEAD REGISTERS
4323
4324  016450  013746  002336                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
4325  016454  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4326                                                                  ;ENABLE INTERRUPT
4327  016460  012677  163506                  MOV     (SP)+,@RHCS1    ;GO WITH
4328                                                                    ;72 IN RHCS1 FOR READ DATA
4329                                                                  ;WITH INTERRUPT ENABLED
4330
4331
4332  016464  104420                          WAT                     ;WAIT FOR IAE BIT TO SET
4333  016466  002174                          RHER1                   ;WAIT FOR RHER1 REGISTER
4334  016470  002000                          IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
4335  016472  000002                          2.                      ;ALLOW 20 MICRO SECONDS
4336  016474  000002                          2.                      ;IAE MUST SET BETWEEN
4337                                                                  ;00 AND 40 MICRO SECONDS
4338
4339                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4340
4341
4342
4343                                          ;AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4344                                          ;RHWC,RHBA,RHCS1,RHCS2,  CANNOT BE PEREDETERMINED
4345                                          ;THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4346  016476  017737  163462  004476          MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
4347                                                                  ;AS EXPLAINED ABOVE
4348  016504  017737  163456  004500          MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
4349                                                                  ;AS EXPLAINED ABOVE
4350  016512  017737  163452  004502          MOV     @RHCS2,@#SAVERE+4;RHCS2 IS UNPREDICTABLE
4351                                                                  ;AS EXPLAINED ABOVE
4352  016520  017737  163446  004504          MOV     @RHCS1,@#SAVERE+6;RHCS1 IS UNPREDICTABLE
4353                                                                  ;AS EXPLAINED ABOVE
4354
4355  016526  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4356  016532  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
4357
4358  016534  000002                          2                       ;2 BIT/BITS TO BE CHANGED
4359  016536  000001                          1                       ;NEW VALUE OF ATA IS 1
4360  016540  100000                          ATA                     ;CHANGE ATA BIT
4361  016542  000001                          1                       ;NEW VALUE OF ERR IS 1
4362  016544  040000                          ERR                     ;CHANGE ERR BIT
4363  016546  017737  163424  004510          MOV     @RHDST,@#SAVERE+12      ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
4364
4365
4366  016554  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4367  016560  002174                          RHER1                   ;CHANGE RHER1 REGISTER
4368
4369  016562  000001                          1                       ;1 BIT/BITS TO BE CHANGED
```

```
4370  016564  000001                          1                       ;NEW VALUE OF IAE IS 1
4371  016566  002000                          IAE                     ;CHANGE IAE BIT
4372
4373  016570  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
4374                                                                    ;FOR WORKING DRIVE IN
4375                                                                    ;SAVED RHAS LOACTION
4376
4377                                          ;COMPARE REGISTERS BEFORE ATTEMPTED READ HEADER
4378                                          ;AND DATA WITH AFTER ATTEMPTED READ
4379
4380  016576  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4381                                                                    ;PRESENT VALUE
4382  016602  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4383  016604  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4384  016606  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4385  016610  016614                          1$                      ;RETURN TO 1$ ON ERROR
4386  016612  016620                          2$                      ;RETURN TO 2$ ON NO ERROR
4387
4388  016614  104054                  1$:     ERROR   54              ;ATTEMPTED READ HEADER
4389  016616  000207                          RTS     PC              ;AND DATA WITH INVALID
4390                                                                    ;ADDRESS CAUSED IMPROPER
4391                                                                    ;REGISTER CHANGE
4392                                                                    ;GOOD DATA GIVES WHAT
4393                                                                    ;SHOULD BE THERE
4394                                                                    ;RECEIVED DATA GIVES
4395                                                                    ;REGISTER CONTENTS
4396                                                                    ;AFTER ATTEMPTED
4397                                                                    ;READ
4398  016620                          2$:
4399
4400
4401                                          ;*****************************************************************
4402                                          ;*TEST 22       INVALID ADDRESS ERROR RHER1 - BIT #10
4403
4404                                          ;*      A READ DATA IS ATTEMPTED TO CYLINDER 0, TRACK 0
4405                                          ;*      SECTOR 20 - FORMAT 18 BITS PER WORD
4406                                          ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
4407                                          ;*      SHOULD SET
4408
4409                                          ;*****************************************************************
4410  016620  000004                  TST22:  SCOPE
4411  016622  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4412  016626  012737  000022  004470          MOV     #22,@#TSTNM     ;SAVE TEST NUMBER
4413
4414  016634  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4415                                                                    ;R3-RHDS1, R4-RHER1
4416                                                                    ;GIVE RH-11 INITIALIZE
4417                                                                    ;SETUP UNIT NUBER
4418
4419                                          ;FILL READ INTO BUFFER WITH 125252
4420  016640  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 260 WORDS, FROM REINTO
4421  016644  003420                          REINTO                  ;STARTING FROM REINTO
4422  016646  000260                          260                     ;260 WORDS
4423  016650  125252                          125252                  ;FILL WITH 125252
4424
4425
```

```
                                        ;THE READ HEADER AND DATA COMMAND IS FILLED
                                        
016652  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
016656  000000                          0                       ;CYLINDER 0
016660     024                  .BYTE   20.                     ;SECTOR 20.
016661     000                  .BYTE   0                       ;TRACK 0
016662  177400                          -256.                   ;WORD COUNT = 256.
016664  003420                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
016666  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
016670  004000                          ECI                     ;18 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
016672  002334                          READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1
                                        
                                        ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
016674  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
016700  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
016702  004476                          SAVEPE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
016704  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.
                                        
016706  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
                                        
                                        
016712  013777  004472  163240          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS
                                        
                                        
016720  013746  002334                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
016724  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
016730  012677  163236                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;70 IN RHCS1 FOR READ DATA
                                                                ;WITH INTERRUPT ENABLED
                                        
                                        
016734  104420                          WAT                     ;WAIT FOR IAE BIT TO SET
016736  002174                          RHER1                   ;WAIT FOR RHER1 REGISTER
016740  002000                          IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
016742  000002                          2.                      ;ALLOW 20 MICRO SECONDS
016744  000002                          2.                      ;IAE MUST SET BETWEEN
                                                                ;00 AND 40 MICRO SECONDS
                                        
                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
                                        
016746  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
016752  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
```

```
4482
4483  016754  000002                        2                       ;2 BIT/BITS TO BE CHANGED
4484  016756  000001                        1                       ;NEW VALUE OF SC IS 1
4485  016760  100000                        SC                      ;CHANGE SC BIT
4486  016762  000001                        1                       ;NEW VALUE OF TRE IS 1
4487  016764  040000                        TRE                     ;CHANGE TRE BIT
4488  016766  017737  163204  004510        MOV     @RHDST,@#SAVERE+12;RHDST IS UNPREDICTABLE
4489
4490  016774  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4491  017000  002214                        RHDS1                   ;CHANGE RHDS1 REGISTER
4492
4493  017002  000002                        2                       ;2 BIT/BITS TO BE CHANGED
4494  017004  000001                        1                       ;NEW VALUE OF ATA IS 1
4495  017006  100000                        ATA                     ;CHANGE ATA BIT
4496  017010  000001                        1                       ;NEW VALUE OF ERR IS 1
4497  017012  040000                        ERR                     ;CHANGE ERR BIT
4498
4499  017014  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4500  017020  002174                        RHER1                   ;CHANGE RHER1 REGISTER
4501
4502  017022  000001                        1                       ;1 BIT/BITS TO BE CHANGED
4503  017024  000001                        1                       ;NEW VALUE OF IAE IS 1
4504  017026  002000                        IAE                     ;CHANGE IAE BIT
4505
4506  017030  053737  004622  004522        BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
4507                                                                ;FOR WORKING DRIVE IN
4508                                                                ;SAVED RHAS LOACTION
4509
4510                                        ;COMPARE REGISTERS BEFORE ATTEMPTED READ
4511                                        ;DATA WITH AFTER ATTEMPTED READ DATA
4512
4513  017036  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4514                                                                ;PRESENT VALUE
4515  017042  004476                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4516  017044  002242                        WC                      ;TEST DATA STARTING FROM 'RHWC'
4517  017046  000022                        18.                     ;18. REGISTERS TO BE COMPARED
4518  017050  017054                        1$                      ;RETURN TO 1$ ON ERROR
4519  017052  017060                        2$                      ;RETURN TO 2$ ON NO ERROR
4520
4521  017054  104054                1$:     ERROR   54              ;ATTEMPTED READ
4522  017056  000207                        RTS     PC              ;DATA WITH INVALID
4523                                                                ;ADDRESS CAUSED IMPROPER
4524                                                                ;REGISTER CHANGE
4525                                                                ;GOOD DATA GIVES WHAT
4526                                                                ;SHOULD BE THERE
4527                                                                ;RECEIVED DATA GIVES
4528                                                                ;REGISTERS CONTENTS
4529                                                                ;AFTER ATTEMPTED
4530                                                                ;READ
4531  017060                        2$:
4532
4533
4534                                ;*****************************************************************
4535                                ;*TEST 23       ADDRESS OVERFLOW ERROR RHER1- BIT#9 AOE
4536
4537                                ;*      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, TRACK 18, SECTOR 21, 261 WORDS
                                ;*
                                ;*      ADDRESS OVERFLOW ERROR RHER1 BIT#9 SHOULD SET
                                ;*      AFTER SECTOR 21 IS WRITTEN
                                ;*      ALL REGISTERS ARE CHECKED
                                ;*
                                ;*      A READ HEADER AND DATA CYLINDER 410, TRACK 18, SECTOR 21,
                                ;*      260+66+4=330  WORDS ARE GIVEN
                                ;*
                                ;*      SECTOR 21 SHOULD BE READ CORRECTLY BUT NO MORE
                                ;*      READS SHOULD HAPPEN, AOE BIT SHOULD SET
                                ;*
                                ;*      CYLINDER 0, TRACK 0, SECTOR 0 IS READ THERE
                                ;*      SHOULD BE NO CHANGE IN DATA IN THIS SECTOR BY
                                ;*      THE LAST WRITE HEADER AND DATA
                                ;*************************************************************
017060  000004                  TST23:  SCOPE
017062  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017066  012737  000023  004470          MOV     #23,@#TSTNM     ;SAVE TEST NUMBER
017074  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                ;FILL WRITE FROM BUFFER WITH HEADER
017100  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
017104  002354                  WRFROM                          ;LOCATION WHERE SAVED
017106  000004                  4                               ;NUMBER OF WORDS SAVED
017110  010000                  10000                           ;FIRST DATA WORD
017112  000000                  0                               ;SECOND DATA WORD
017114  000000                  0                               ;THIRD DATA WORD
017116  000000                  0                               ;FOURTH DATA WORD
                                ;FILL WRITE FROM BUFFER WITH DATA
017120  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
017124  002364                  WRFROM+<4*2>                            ;STARTING FROM WRFROM+<4*2>
017126  000400                  256.                            ;256. WORDS
017130  000000                  0                               ;FILL WITH 0
                                ;FILL   WRITE HEADER AND DATA COMMAND
017132  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
017136  000000                  0                               ;CYLINDER 0
017140     000          .BYTE   0                               ;SECTOR 0
017141     000          .BYTE   0                               ;TRACK 0
017142  177374                  -256.-4         ;WORD COUNT (DATA)=256.+
                                                                ;4 HEADER WORDS
017144  002354                  WRFROM                          ;BUS ADDRESS
```

```
4594                                                    ;STARTING ADDRESS OF DATA
4595                                                    ;BUFFER = WRFROM
4596  017146  000000                    0               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4597  017150  010000                    FMT22           ;16 BITS PER WORD FORMAT
4598                                                    ;DO NOT INHIBIT ECC CORRECTION
4599                                                    ;DO NOT INHIBIT HEADER COMPARE
4600  017152  002332                    WRIFOR          ;GET READY TO DO A WRIFOR
4601                                              ;WRITE HEADER AND DATA WITH 62 IN RHCS1
4602
4603
4604  017154  004737  030140            JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
4605
4606
4607  017160  013777  004472  162772    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4608                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
4609                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4610                                                    ;'TIME' WILL ONLY SAVE
4611                                                    ;CURRENT CYLINDER ADDRESS
4612                                                    ;AND LOOK AHEAD REGISTERS
4613
4614
4615  017166  013746  002332            MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
4616  017172  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4617                                                    ;ENABLE INTERRUPT
4618  017176  012677  162770            MOV     (SP)+,@RHCS1    ;GO WITH
4619                                                     ;62 IN RHCS1 FOR WRITE HEADER AND DATA
4620                                                    ;WITH INTERRUPT ENABLED
4621
4622
4623  017202  104420                    WAT                     ;WAIT FOR RDY BIT TO SET
4624  017204  002172                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
4625  017206  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4626  017210  004704                    2500.                   ;ALLOW 25000 MICRO SECONDS
4627  017212  004704                    2500.                   ;RDY MUST SET BETWEEN
4628                                                    ;00 AND 50000 MICRO SECONDS
4629
4630  017214  004737  030060            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4631                                                    ;R3-RHDS1, R4-RHER1
4632                                                    ;GIVE RH-11 INITIALIZE
4633                                                    ;SETUP UNIT NUBER
4634                                    ;FILL WRITE FROM BUFFER WITH HEADER
4635
4636  017220  004037  027704            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
4637  017224  002354                    WRFROM                  ;LOCATION WHERE SAVED
4638  017226  000004                    4                       ;NUMBER OF WORDS SAVED
4639  017230  010632                    10632                   ;FIRST DATA WORD
4640  017232  011025                    <18.*400>!<21.>                 ;SECOND DATA WORD
4641  017234  000000                    0                       ;THIRD DATA WORD
4642  017236  000000                    0                       ;FOURTH DATA WORD
4643
4644                                    ;FILL WRITE FROM BUFFER WITH DATA - 65125
4645  017240  004037  027730            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4646  017244  002364                    WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
4647  017246  000400                    256.                    ;256. WORDS
4648  017250  065125                    <26.*2000>!<18.*40>!<21.>                       ;FILL WITH <26.*2000>!<18.*40>!<
4649
```

```
4650
4651                                    ;FILL WRITE FROM BUFFER WITH NEXT HEADR
4652                                    ;THIS IS A NON EXISTANT HEADER AND SHOULD NOT BE WRITTEN
4653                                    ;BY AOE SHUTTING THE WRITE
4654
4655  017252  004037  027704            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
4656  017256  003364                    WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
4657  017260  000004                    4                       ;NUMBER OF WORDS SAVED
4658  017262  010633                    10633                   ;FIRST DATA WORD
4659  017264  000000                    0                       ;SECOND DATA WORD
4660  017266  000000                    0                       ;THIRD DATA WORD
4661  017270  000000                    0                       ;FOURTH DATA WORD
4662
4663                                    ;FILL WRITE FROM BUFFER WITH DATA FOR NEXT SECTOR
4664  017272  004037  027730            JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
4665  017276  003374                    WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
4666  017300  000002                    2                       ;2 WORDS
4667  017302  066000                    <27.*2000>                      ;FILL WITH <27.*2000>
4668
4669
4670                                    ;FILL WRITE HEADER AND DATA COMMAND
4671
4672  017304  004037  032024            JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
4673  017310  000632                    410.                    ;CYLINDER 410.
4674  017312     025            .BYTE   21.                     ;SECTOR 21.
4675  017313     022            .BYTE   18.                     ;TRACK 18.
4676  017314  177373                    -257.-4         ;WORD COUNT (DATA)=257.+
4677                                                            ;4 HEADER WORDS
4678  017316  002354                    WRFROM                  ;BUS ADDRESS
4679                                                            ;STARTING ADDRESS OF DATA
4680                                                            ;BUFFER = WRFROM
4681  017320  000000                    0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4682  017322  010000                    FMT22                   ;16 BITS PER WORD FORMAT
4683                                                            ;DO NOT INHIBIT ECC CORRECTION
4684                                                            ;DO NOT INHIBIT HEADER COMPARE
4685  017324  002332                    WRIFOR                  ;GET READY TO DO A WRIFOR
4686                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
4687
4688
4689                                    ;SAVE REGISTERS FOR COMPARISON AFTER WIRTE HEADER AND DATA
4690  017326  004037  030220            JSR     R0,@#SAVER      ;SAVE REGISTERS
4691  017332  002164                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4692  017334  004476                    SAVERE                  ;STARTING ADDRES OF WHERE
4693                                                            ;THE REGISTERS ARE SAVED
4694  017336  000022                    18.                     ;NUMBER OF REGISTERS
4695                                                            ;SAVED = 18.
4696
4697  017340  004737  030140            JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
4698
4699
4700  017344  013777  004472  162606    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4701                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
4702                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4703                                                            ;'TIME' WILL ONLY SAVE
4704                                                            ;CURRENT CYLINDER ADDRESS
4705                                                            ;AND LOOK AHEAD REGISTERS
```

```
4706
4707
4708  017352  013746  002332                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
4709  017356  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4710                                                                    ;ENABLE INTERRUPT
4711  017362  012677  162604                  MOV     (SP)+,@RHCS1    ;GO WITH
4712                                                                     ;62 IN RHCS1 FOR WRITE HEADER AND DATA
4713                                                                    ;WITH INTERRUPT ENABLED
4714
4715
4716  017366  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
4717  017370  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
4718  017372  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4719  017374  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
4720  017376  004704                          2500.                   ;RDY MUST SET BETWEEN
4721                                                                    ;00 AND 50000 MICRO SECONDS
4722
4723                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4724  017400  004037  027762                  JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2>+<1.*2> INTO SAVED RHBA
4725  017404  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
4726  017406  003366                          WRFROM+<260.*2>+<1.*2>                  ;DATA
4727  017410  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
4728  017414  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
4729  017416  000000                          0                       ;DATA
4730
4731  017420  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4732  017424  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
4733
4734  017426  000002                          2                       ;2 BIT/BITS TO BE CHANGED
4735  017430  000001                          1                       ;NEW VALUE OF SC IS 1
4736  017432  100000                          SC                      ;CHANGE SC BIT
4737  017434  000001                          1                       ;NEW VALUE OF TRE IS 1
4738  017436  040000                          TRE                     ;CHANGE TRE BIT
4739
4740  017440  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4741  017444  002170                          RHCS2                   ;CHANGE RHCS2 REGISTER
4742
4743  017446  000002                          2                       ;2 BIT/BITS TO BE CHANGED
4744  017450  000001                          1                       ;NEW VALUE OF OR IS 1
4745  017452  000200                          OR                      ;CHANGE OR BIT
4746  017454  000001                          1                       ;NEW VALUE OF IR IS 1
4747  017456  000100                          IR                      ;CHANGE IR BIT
4748  017460  004037  027762                  JSR     R0,@#FILLRE     ;MOV AOE INTO SAVED RHER1
4749  017464  002174                          RHER1                   ;SAVED REGISTER TO CHANGE
4750  017466  001000                          AOE                     ;DATA
4751
4752  017470  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4753  017474  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
4754
4755  017476  000003                          3                       ;3 BIT/BITS TO BE CHANGED
4756  017500  000001                          1                       ;NEW VALUE OF ATA IS 1
4757  017502  100000                          ATA                     ;CHANGE ATA BIT
4758  017504  000001                          1                       ;NEW VALUE OF ERR IS 1
4759  017506  040000                          ERR                     ;CHANGE ERR BIT
4760  017510  000001                          1                       ;NEW VALUE OF LBT IS 1
4761  017512  002000                          LBT                     ;CHANGE LBT BIT
```

```
4762
4763  017514  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
4764                                                                          ;FOR WORKING DRIVE IN
4765                                                                          ;SAVED RHAS LOACTION
4766  017522  004037  027762                  JSR     R0,@#FILLRE     ;MOV 633 INTO SAVED RHCA
4767  017526  002204                          RHCA                    ;SAVED REGISTER TO CHANGE
4768  017530  000633                          633                     ;DATA
4769  017532  004037  027762                  JSR     R0,@#FILLRE     ;MOV 632 INTO SAVED RHCC
4770  017536  002226                          RHCC                    ;SAVED REGISTER TO CHANGE
4771  017540  000632                          632                     ;DATA
4772  017542  017737  162430  004510          MOV     @RHDST,@#SAVERE+12      ;RHDST IS UNPREDICTABLE
4773
4774                                          ;COMPARE REGISTERS BEFORE WRITE HEADER AND DATA WITH AFTER
4775
4776  017550  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4777                                                                  ;PRESENT VALUE
4778  017554  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4779  017556  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4780  017560  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4781  017562  017566                          1$                      ;RETURN TO 1$ ON ERROR
4782  017564  017572                          2$                      ;RETURN TO 2$ ON NO ERROR
4783
4784  017566  104055                  1$:     ERROR   55              ;WRITING HEADER AND DATA WITH
4785  017570  000207                          RTS     PC              ;EXPECTED ADDRESS OVERFLOW ERROR
4786                                                                  ;CAUSED IMPROPER REGISTER
4787                                                                  ;CHANGE
4788                                                                  ;GOOD DATA GIVES WHAT SHOULD
4789                                                                  ;BE THERE
4790                                                                  ;RECEIVED DATA GIVES WHAT
4791                                                                  ;WAS THERE AFTER WRITE
4792                                                                  ;HEADER AND DATA
4793
4794                                          ;NOW PREPARE TO DO A READ HEADER AND DATA
4795
4796                                          ;FILL WRITE FROM BUFFER WITH EXPECTED HEADER
4797  017572                          2$:
4798
4799  017572  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4800                                                                  ;R3-RHDS1, R4-RHER1
4801                                                                  ;GIVE RH-11 INITIALIZE
4802                                                                  ;SETUP UNIT NUBER
4803
4804  017576  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
4805  017602  002354                          WRFROM                  ;LOCATION WHERE SAVED
4806  017604  000004                          4                       ;NUMBER OF WORDS SAVED
4807  017606  010632                          10632                   ;FIRST DATA WORD
4808  017610  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
4809  017612  000000                          0                       ;THIRD DATA WORD
4810  017614  000000                          0                       ;FOURTH DATA WORD
4811
4812                                          ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
4813  017616  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
4814  017622  002364                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
4815  017624  000400                          256.                    ;256. WORDS
4816  017626  065125                          <26.*2000>!<18.*40>!<21.>                       ;FILL WITH <26.*2000>!<18.*40>!<
4817
```

```
4818
4819                                    ;FILL WRITE FROM BUFFER WITH 377 FRO WORDS 261 TO 266
4820  017630  004037  027730             JSR     R0,@#CLAREA     ;CLEAR 6 WORDS, FROM WRFROM+<260.*2>
4821  017634  003364                     WRFROM+<260.*2>                 ;STARTING FROM WRFROM+<260.*2>
4822  017636  000006                     6                       ;6 WORDS
4823  017640  000377                     377                     ;FILL WITH 377
4824
4825
4826                                    ;CLEAR READ INTO BUFFER
4827  017642  004037  027730             JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM REINTO
4828  017646  003420                     REINTO                  ;STARTING FROM REINTO
4829  017650  000412                     266.                    ;266. WORDS
4830  017652  000377                     377                     ;FILL WITH 377
4831
4832
4833
4834  017654  004737  030060             JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4835                                                             ;R3-RHDS1, R4-RHER1
4836                                                             ;GIVE RH-11 INITIALIZE
4837                                                             ;SETUP UNIT NUBER
4838
4839                                    ;FILL READ HEADER AND DATA COMMAND
4840
4841  017660  004037  032024             JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
4842  017664  000632                     410.                    ;CYLINDER 410.
4843  017666     025              .BYTE   21.                     ;SECTOR 21.
4844  017667     022              .BYTE   18.                     ;TRACK 18.
4845  017670  177266                     -326.-4         ;WORD COUNT (DATA)=326.+
4846                                                             ;4 HEADER WORDS
4847  017672  003420                     REINTO                  ;BUS ADDRESS
4848                                                             ;STARTING ADDRESS OF DATA
4849                                                             ;BUFFER = REINTO
4850  017674  000000                     0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4851  017676  014000                     ECI!FMT22                       ;16 BITS PER WORD FORMAT
4852                                                             ;INHIBIT ECC CORRECTION
4853                                                             ;DO NOT INHIBIT HEADER COMPARE
4854  017700  002336                     REFOR                   ;GET READY TO DO A REFOP
4855                                                             ;READ HEADER AND DATA WITH 72 IN RHCS1
4856
4857
4858                                    ;SAVE REGISTERS FOR COMPARISON AFTER
4859                                    ;READ HEADER AND DATA
4860  017702  004037  030220             JSR     R0,@#SAVER      ;SAVE REGISTERS
4861  017706  002164                     RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4862  017710  004476                     SAVERE                  ;STARTING ADDRES OF WHERE
4863                                                             ;THE REGISTERS ARE SAVED
4864  017712  000022                     18.                     ;NUMBER OF REGISTERS
4865                                                             ;SAVED = 18.
4866
4867  017714  004737  030140             JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
4868
4869
4870  017720  013777  004472  162232     MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4871                                                             ;TO 'TIME1' IF P-CLOCK IS PRESENT
4872                                                             ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4873                                                             ;'TIME' WILL ONLY SAVE
```

```
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

017726  013746  002336          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
017732  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
017736  012677  162230          MOV     (SP)+,@RHCS1    ;GO WITH
                                                          ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED

017742  104420                  WAT                     ;WAIT FOR RDY BIT TO SET
017744  002172                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
017746  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
017750  001614                  900.                    ;ALLOW 9000 MICRO SECONDS
017752  001507                  839.                    ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS

                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
017754  004037  027762          JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
017760  002166                  RHBA                    ;SAVED REGISTER TO CHANGE
017762  004430                  REINTO+<260.*2>                 ;DATA
017764  004037  027762          JSR     R0,@#FILLRE     ;MOV -70. INTO SAVED RHWC
017770  002164                  RHWC                    ;SAVED REGISTER TO CHANGE
017772  177672                  -70.                    ;DATA

017774  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020000  002172                  RHCS1                   ;CHANGE RHCS1 REGISTER

020002  000002                  2                       ;2 BIT/BITS TO BE CHANGED
020004  000001                  1                       ;NEW VALUE OF SC IS 1
020006  100000                  SC                      ;CHANGE SC BIT
020010  000001                  1                       ;NEW VALUE OF TRE IS 1
020012  040000                  TRE                     ;CHANGE TRE BIT

020014  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020020  002214                  RHDS1                   ;CHANGE RHDS1 REGISTER

020022  000003                  3                       ;3 BIT/BITS TO BE CHANGED
020024  000001                  1                       ;NEW VALUE OF ATA IS 1
020026  100000                  ATA                     ;CHANGE ATA BIT
020030  000001                  1                       ;NEW VALUE OF ERR IS 1
020032  040000                  ERR                     ;CHANGE ERR BIT
020034  000001                  1                       ;NEW VALUE OF LBT IS 1
020036  002000                  LBT                     ;CHANGE LBT BIT
020040  004037  027762          JSR     R0,@#FILLRE     ;MOV AOE INTO SAVED RHER1
020044  002174                  RHER1                   ;SAVED REGISTER TO CHANGE
020046  001000                  AOE                     ;DATA
020050  004037  027762          JSR     R0,@#FILLRE     ;MOV 633 INTO SAVED RHCA
020054  002204                  RHCA                    ;SAVED REGISTER TO CHANGE
020056  000633                  633                     ;DATA

020060  053737  004622  004522  BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
                                                                ;FOR WORKING DRIVE IN
                                                                ;SAVED RHAS LOACTION
```

```
 4930  020066  017737  162104  004510          MOV     @RHDST,@#SAVERE+12      ;RHDST IS UNPREDICTABLE
 4931
 4932                                          ;COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
 4933                                          ;REGISTERS AFTER COMMAND
 4934
 4935  020074  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 4936                                                                  ;PRESENT VALUE
 4937  020100  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 4938  020102  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 4939  020104  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 4940  020106  020112                          3$                      ;RETURN TO 3$ ON ERROR
 4941  020110  020116                          4$                      ;RETURN TO 4$ ON NO ERROR
 4942
 4943  020112  104055                  3$:     ERROR   55              ;READING HEADER AND DATA WITH
 4944  020114  000207                          RTS     PC              ;EXPECTED ADDRESS OVERFLOW
 4945                                                                  ;ERROR CAUSED IMPROPER
 4946                                                                  ;REGISTER CHANGE
 4947                                                                  ;GOOD DATA GIVES WHAT SHOULD
 4948                                                                  ;BE THERE
 4949                                                                  ;RECEIVED DATA GIVES WHAT
 4950                                                                  ;WAS THERE AFTER COMMAND
 4951
 4952                                          ;NOW COMPARE THE DATA READ
 4953  020116                          4$:
 4954
 4955  020116  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 4956  020122  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
 4957  020124  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
 4958  020126  000412                          266.                    ;266. WORDS TO BE COMPARED
 4959  020130  020134                          5$                      ;RETURN TO 5$ ON ERROR
 4960  020132  020140                          6$                      ;RETURN TO 6$ ON NO ERROR
 4961
 4962
 4963  020134  104056                  5$:     ERROR   56              ;DATA READ WITH AN EXPECTED
 4964  020136  000207                          RTS     PC              ;ADDRESS OVERFLOW ERROR
 4965                                                                  ;IS INCORRECT
 4966                                                                  ;WORD NO 1 TO 260 SHOULD
 4967                                                                  ;BE READ CORRECTLY
 4968                                                                  ;WORD NO 261 TO 266 SHOULD
 4969                                                                  ;NOT CHANGE DUE TO THE READ
 4970  020140                          6$:
 4971
 4972  020140  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 4973                                                                  ;R3-RHDS1, R4-RHER1
 4974                                                                  ;GIVE RH-11 INITIALIZE
 4975                                                                  ;SETUP UNIT NUBER
 4976                                          ;NOW PREPARE TO READ CYLINDER 0, SECTOR 0, TRACK 0
 4977                                          ;TO SEE THAT NOTHING GOT WRITTEN ON THERE
 4978                                          ;WITH THE ADDRESS OVER FLOW
 4979
 4980                                          ;FILL WRITE FROM BUFFER WITH EXPECTED HEADER
 4981
 4982  020144  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
 4983  020150  002354                          WRFROM                  ;LOCATION WHERE SAVED
 4984  020152  000004                          4                       ;NUMBER OF WORDS SAVED
 4985  020154  010000                          10000                   ;FIRST DATA WORD
```

```
 4986  020156  000000                          0                       ;SECOND DATA WORD
 4987  020160  000000                          0                       ;THIRD DATA WORD
 4988  020162  000000                          0                       ;FOURTH DATA WORD
 4989  020164  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
 4990  020170  002364                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
 4991  020172  000400                          256.                    ;256. WORDS
 4992  020174  000000                          0                       ;FILL WITH 0
 4993
 4994
 4995                                          ;FILL READ INTO BUFFER WITH 377
 4996  020176  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
 4997  020202  003420                          REINTO                  ;STARTING FROM REINTO
 4998  020204  000404                          260.                    ;260. WORDS
 4999  020206  000377                          377                     ;FILL WITH 377
 5000
 5001
 5002
 5003  020210  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 5004                                                                  ;R3-RHDS1, R4-RHER1
 5005                                                                  ;GIVE RH-11 INITIALIZE
 5006                                                                  ;SETUP UNIT NUBER
 5007
 5008                                          ;FILL COMMAND FOR READ HEADER AND DATA
 5009
 5010  020214  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 5011  020220  000000                          0                       ;CYLINDER 0
 5012  020222     000                  .BYTE   0                       ;SECTOR 0
 5013  020223     000                  .BYTE   0                       ;TRACK 0
 5014  020224  177374                          -256.-4         ;WORD COUNT (DATA)=256.+
 5015                                                                  ;4 HEADER WORDS
 5016  020226  003420                          REINTO                  ;BUS ADDRESS
 5017                                                                  ;STARTING ADDRESS OF DATA
 5018                                                                  ;BUFFER = REINTO
 5019  020230  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 5020  020232  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
 5021                                                                  ;INHIBIT ECC CORRECTION
 5022                                                                  ;DO NOT INHIBIT HEADER COMPARE
 5023  020234  002336                          REFOR                   ;GET READY TO DO A REFOR
 5024                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
 5025
 5026
 5027                                          ;SAVE REGISTERS FOR COMPARISON AFTER READ
 5028  020236  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
 5029  020242  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 5030  020244  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
 5031                                                                  ;THE REGISTERS ARE SAVED
 5032  020246  000021                          17.                     ;NUMBER OF REGISTERS
 5033                                                                  ;SAVED = 17.
 5034  020250  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
 5035
 5036
 5037  020254  013777  004472  161676          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 5038                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 5039                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 5040                                                                  ;'TIME' WILL ONLY SAVE
 5041                                                                  ;CURRENT CYLINDER ADDRESS
```

```
5042                                                    ;AND LOOK AHEAD REGISTERS
5043
5044
5045  020262  013746  002336          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
5046  020266  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5047                                                    ;ENABLE INTERRUPT
5048  020272  012677  161674          MOV     (SP)+,@RHCS1    ;GO WITH
5049                                                      ;72 IN RHCS1 FOR READ DATA
5050                                                    ;WITH INTERRUPT ENABLED
5051
5052
5053  020276  104420                  WAT                     ;WAIT FOR RDY BIT TO SET
5054  020300  002172                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
5055  020302  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5056  020304  004704                  2500.                   ;ALLOW 25000 MICRO SECONDS
5057  020306  004704                  2500.                   ;RDY MUST SET BETWEEN
5058                                                    ;00 AND 50000 MICRO SECONDS
5059
5060                                  ;CHANGE REGISTERS TO EXPECTED VALUE
5061  020310  004037  027762          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
5062  020314  002164                  RHWC                    ;SAVED REGISTER TO CHANGE
5063  020316  000000                  0                       ;DATA
5064  020320  004037  027762          JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
5065  020324  002166                  RHBA                    ;SAVED REGISTER TO CHANGE
5066  020326  004430                  REINTO+<260.*2>                 ;DATA
5067  020330  004037  027762          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5068  020334  002176                  RHDST                   ;SAVED REGISTER TO CHANGE
5069  020336  000001                  1                       ;DATA
5070
5071  020340                  ST22A:  ;COMPARE REGISTER BEFORE READ WITH AFTER
5072
5073  020340  004037  031040          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5074                                                    ;PRESENT VALUE
5075  020344  004476                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5076  020346  002242                  WC                      ;TEST DATA STARTING FROM 'RHWC'
5077  020350  000021                  17.                     ;17. REGISTERS TO BE COMPARED
5078  020352  020356                  4$                      ;RETURN TO 4$ ON ERROR
5079  020354  020362                  1$                      ;RETURN TO 1$ ON NO ERROR
5080
5081  020356  104031          4$:     ERROR   31              ;READ HEADER AND DATA ON
5082  020360  000207                  RTS     PC              ;CYLINDER 0, SECTOR 0
5083                                                    ;TRACK 0 AFTER A FORCED
5084                                                    ;ADDRESS OVER FLOW ERROR
5085                                                    ;CAUSED IMPROPER REGISTER
5086                                                    ;CHANGE
5087                                                    ;GOOD DATA GIVES WHAT
5088                                                    ;SHOULD BE THERE
5089                                                    ;RECEIVED DATA GIVES WHAT
5090                                                    ;WAS THERE AFTER READ
5091                                                    ;HEADER AND DATA
5092                                                    ;IF HEADER COMPARE ERROR
5093                                                    ;IS FOUND AND THE DATA
5094                                                    ;ERROR GIVES THE NEW
5095                                                    ;HEADER TO
5096                                                    ;CYLINDER 633 (OCTAL)
5097                                                    ;THEN AOE OVER FLOWED
```

```
                                                        ;INTO HERE
                                                        ;
                                    ;COMPARE DAT READ
020362                          1$:
                                                        ;
020362  004037  032070              JSR     R0,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
020366  002354                      WRFROM              ;GOOD DATA      STARTS FROM WRFROM
020370  003420                      REINTO              ;TEST DATA STARTS FROM REINTO
020372  000404                      260.                ;260. WORDS TO BE COMPARED
020374  020400                      2$                  ;RETURN TO 2$ ON ERROR
020376  020404                      3$                  ;RETURN TO 3$ ON NO ERROR
                                                        ;
                                                        ;
020400  104032                  2$: ERROR   32          ;READ HEADER AND DATA
020402  000207                      RTS     PC          ;ON CYLINDER 0, TRACK 0
                                                        ;SECTOR 0 AFTER A FORCED
                                                        ;'AOE' ERROR CAUSED
                                                        ;AN ERROR
                                                        ;IF FIRST WORD IS
                                                        ;10633 (OCTAL) THEN
                                                        ;AOE OVER FLOWED INTO HERE
020404                          3$:
                                                        ;
                                                        ;
                                                        ;
                                ;*******************************************************************
                                ;*TEST 24        FORMAT ERROR RHER1 BIT #4
                                                        ;
                                ;*      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 FOR BIT #4 IN RHER1
                                ;*
                                ;*      THEN THIS SECTOR WILL BE READ IN THE CORRECT FORMAT
                                ;*      16 BITS PER WORD TO CHECK THAT NOTHING GOT WRITTEN
                                ;*******************************************************************
020404  000004                  TST24:  SCOPE
020406  012706  001000              MOV     #STACK,SP   ;RESET STACK
020412  012737  000024  004470      MOV     #24,@#TSTNM ;SAVE TEST NUMBER
                                                        ;
020420  004737  030060              JSR     PC,@#CLDISK ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;
                                    ;FIRST WRITE HEADER AND DATA CYLINDER 0, TRACK 0, SECTOR 0
                                    ;FILL WRITE FROM BUFFER WITH HEADER
                                                        ;
020424  004037  027704              JSR     R0,@#FLHEAD ;SAVE HEADER DATA IN WRFROM
020430  002354                      WRFROM              ;LOCATION WHERE SAVED
020432  000004                      4                   ;NUMBER OF WORDS SAVED
020434  010000                      10000               ;FIRST DATA WORD
020436  000000                      0                   ;SECOND DATA WORD
020440  000000                      0                   ;THIRD DATA WORD
```

```
5154  020442  000000                          0                       ;FOURTH DATA WORD
5155                                          ;FILL WRITE FROM BUFFER WITH DATA
5156  020444  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
5157  020450  002364                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
5158  020452  000400                          256.                    ;256. WORDS
5159  020454  000000                          0                       ;FILL WITH 0
5160
5161
5162                                          ;FILL COMMAND
5163
5164  020456  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5165  020462  000000                          0                       ;CYLINDER 0
5166  020464     000                  .BYTE   0                       ;SECTOR 0
5167  020465     000                  .BYTE   0                       ;TRACK 0
5168  020466  177374                          -256.-4         ;WORD COUNT (DATA)=256.+
5169                                                                  ;4 HEADER WORDS
5170  020470  002354                          WRFROM                  ;BUS ADDRESS
5171                                                                  ;STARTING ADDRESS OF DATA
5172                                                                  ;BUFFER = WRFROM
5173  020472  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5174  020474  010000                          FMT22                   ;16 BITS PER WORD FORMAT
5175                                                                  ;DO NOT INHIBIT ECC CORRECTION
5176                                                                  ;DO NOT INHIBIT HEADER COMPARE
5177  020476  002332                          WRIFOR                  ;GET READY TO DO A WRIFOR
5178                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
5179
5180
5181  020500  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5182
5183
5184  020504  013777  004472  161446          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5185                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5186                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5187                                                                  ;'TIME' WILL ONLY SAVE
5188                                                                  ;CURRENT CYLINDER ADDRESS
5189                                                                  ;AND LOOK AHEAD REGISTERS
5190
5191
5192  020512  013746  002332                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
5193  020516  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5194                                                                  ;ENABLE INTERRUPT
5195  020522  012677  161444                  MOV     (SP)+,@RHCS1    ;GO WITH
5196                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
5197                                                                  ;WITH INTERRUPT ENABLED
5198
5199
5200  020526  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
5201  020530  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
5202  020532  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5203  020534  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
5204  020536  004704                          2500.                   ;RDY MUST SET BETWEEN
5205                                                                  ;00 AND 50000 MICRO SECONDS
5206
5207  020540  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5208                                                                  ;R3-RHDS1, R4-RHER1
5209                                                                  ;GIVE RH-11 INITIALIZE
```

```
5210                                                    ;SETUP UNIT NUBER
5211                                    ;NOW PREPARE TO WRITE WITH WRONG FORMAT
5212
5213                                    ;FILL WRITE FROM BUFFER
5214  020544  004037  027730              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
5215  020550  002354                      WRFROM                  ;STARTING FROM WRFROM
5216  020552  000400                      256.                    ;256. WORDS
5217  020554  000377                      377                     ;FILL WITH 377
5218
5219
5220                                    ;FILL WRITE DATA COMMAND
5221
5222  020556  004037  032024              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5223  020562  000000                      0                       ;CYLINDER 0
5224  020564     000              .BYTE   0                       ;SECTOR 0
5225  020565     000              .BYTE   0                       ;TRACK 0
5226  020566  177400                      -256.                   ;WORD COUNT = 256.
5227  020570  002354                      WRFROM                  ;BUS ADDRESS
5228                                                            ;STARTING ADDRESS OF DATA
5229                                                            ;BUFFER = WRFROM
5230  020572  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5231  020574  000000                      0                       ;18 BITS PER WORD FORMAT
5232                                                            ;DO NOT INHIBIT ECC CORRECTION
5233                                                            ;DO NOT INHIBIT HEADER COMPARE
5234  020576  002330                      WRIDAT                  ;GET READY TO DO A WRIDAT
5235                                                  ;WRITE DATA WITH 60 IN RHCS1
5236
5237                                    ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
5238                                    ;WITH WRONG FORMAT
5239  020600  004037  030220              JSR     R0,@#SAVER      ;SAVE REGISTERS
5240  020604  002164                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5241  020606  004476                      SAVERE                  ;STARTING ADDRES OF WHERE
5242                                                            ;THE REGISTERS ARE SAVED
5243  020610  000022                      18.                     ;NUMBER OF REGISTERS
5244                                                            ;SAVED = 18.
5245
5246  020612  004737  030140              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5247
5248
5249  020616  013777  004472  161334      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5250                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
5251                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5252                                                            ;'TIME' WILL ONLY SAVE
5253                                                            ;CURRENT CYLINDER ADDRESS
5254                                                            ;AND LOOK AHEAD REGISTERS
5255
5256
5257  020624  013746  002330              MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
5258  020630  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5259                                                            ;ENABLE INTERRUPT
5260  020634  012677  161332              MOV     (SP)+,@RHCS1    ;GO WITH
5261                                                             ;60 IN RHCS1 FOR WRITE DATA
5262                                                            ;WITH INTERRUPT ENABLED
5263
5264
5265  020640  104420                      WAT                     ;WAIT FOR RDY BIT TO SET
```

```
5266  020642  002172                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
5267  020644  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5268  020646  001522                  850.                    ;ALLOW 8500 MICRO SECONDS
5269  020650  001510                  840.                    ;RDY MUST SET BETWEEN
5270                                                          ;100 AND 16900 MICRO SECONDS
5271
5272                                  ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
5273  020652  004037  027762          JSR     R0,@#FILLRE     ;MOV -248. INTO SAVED RHWC
5274  020656  002164                  RHWC                    ;SAVED REGISTER TO CHANGE
5275  020660  177410                  -248.                   ;DATA
5276  020662  004037  027762          JSR     R0,@#FILLRE     ;MOV WRFROM+<8.*2> INTO SAVED RHBA
5277  020666  002166                  RHBA                    ;SAVED REGISTER TO CHANGE
5278  020670  002374                  WRFROM+<8.*2>                   ;DATA
5279
5280  020672  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5281  020676  002172                  RHCS1                   ;CHANGE RHCS1 REGISTER
5282
5283  020700  000002                  2                       ;2 BIT/BITS TO BE CHANGED
5284  020702  000001                  1                       ;NEW VALUE OF SC IS 1
5285  020704  100000                  SC                      ;CHANGE SC BIT
5286  020706  000001                  1                       ;NEW VALUE OF TRE IS 1
5287  020710  040000                  TRE                     ;CHANGE TRE BIT
5288
5289  020712  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5290  020716  002170                  RHCS2                   ;CHANGE RHCS2 REGISTER
5291
5292  020720  000001                  1                       ;1 BIT/BITS TO BE CHANGED
5293  020722  000001                  1                       ;NEW VALUE OF OR IS 1
5294  020724  000200                  OR                      ;CHANGE OR BIT
5295
5296  020726  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5297  020732  002214                  RHDS1                   ;CHANGE RHDS1 REGISTER
5298
5299  020734  000002                  2                       ;2 BIT/BITS TO BE CHANGED
5300  020736  000001                  1                       ;NEW VALUE OF ATA IS 1
5301  020740  100000                  ATA                     ;CHANGE ATA BIT
5302  020742  000001                  1                       ;NEW VALUE OF ERR IS 1
5303  020744  040000                  ERR                     ;CHANGE ERR BIT
5304  020746  004037  027762          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5305  020752  002176                  RHDST                   ;SAVED REGISTER TO CHANGE
5306  020754  000001                  1                       ;DATA
5307
5308  020756  053737  004622  004522  BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
5309                                                          ;FOR WORKING DRIVE IN
5310                                                          ;SAVED RHAS LOACTION
5311
5312  020764  004037  030732          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5313  020770  002174                  RHER1                   ;CHANGE RHER1 REGISTER
5314
5315  020772  000001                  1                       ;1 BIT/BITS TO BE CHANGED
5316  020774  000001                  1                       ;NEW VALUE OF FER IS 1
5317  020776  000020                  FER                     ;CHANGE FER BIT
5318  021000  017746  161164          MOV     @RHCS2,-(SP)    ;GET RHCS2
5319  021004  042716  177477          BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
5320  021010  042737  000300  004502  BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
5321  021016  052637  004502          BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
```

```
                                        ;COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT

021022  004037  031040               JSR     P0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                             ;PRESENT VALUE
021026  004476                       SAVERF                  ;GOOD DATA SAVED IN 'SAVERE'
021030  002242                       WC                      ;TEST DATA STARTING FROM 'RHWC'
021032  000022                       18.                     ;18. REGISTERS TO BE COMPARED
021034  021040                       1$                      ;RETURN TO 1$ ON ERROR
021036  021044                       2$                      ;RETURN TO 2$ ON NO ERROR

021040  104057              1$:      ERROR   57              ;ATTEMPTING TO WRITE DATA
021042  000207                       RTS     PC              ;WITH WRONG FORMAT BIT CAUSED
                                                             ;IMPROPER REGISTER CHANGE
                                                             ;GOOD DATA GIVES WHAT SHOULD
                                                             ;BE THERE
                                                             ;RECEIVED DATA GIVES WHAT WAS
                                                             ;THERE AFTER ATTEMPTED WRITE

                                        ;NOW PREPARE TO READ WITH CORRECT FORMAT TO CHECK
                                        ;THAT NOTHING GOT WRITTEN
021044                          2$:

021044  004737  030060               JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                             ;R3-RHDS1, R4-RHER1
                                                             ;GIVE RH-11 INITIALIZE
                                                             ;SETUP UNIT NUBER
                                        ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
021050  004037  027730               JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
021054  002354                       WRFROM                  ;STARTING FROM WRFROM
021056  000400                       256.                    ;256. WORDS
021060  000000                       0                       ;FILL WITH 0

                                        ;FILL READ INTO BUFFER WITH 125252
021062  004037  027730               JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
021066  003420                       REINTO                  ;STARTING FROM REINTO
021070  000400                       256.                    ;256. WORDS
021072  125252                       125252                  ;FILL WITH 125252

                                        ;FILL COMMAND TO READ DATA

021074  004037  032024               JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
021100  000000                       0                       ;CYLINDER 0
021102     000              .BYTE    0                       ;SECTOR 0
021103     000              .BYTE    0                       ;TRACK 0
021104  177400                       -256.                   ;WORD COUNT = 256.
021106  003420                       REINTO                  ;BUS ADDRESS
                                                             ;STARTING ADDRESS OF DATA
                                                             ;BUFFER = REINTO
021110  000000                       0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
021112  014000                       FMT22!ECI                       ;16 BITS PER WORD FORMAT
                                                             ;INHIBIT ECC CORRECTION
```

```
                                                        ;DO NOT INHIBIT HEADER COMPARE
021114  002334                          READAT          ;GET READY TO DO A READAT
                                                ;READ DATA WITH 70 IN RHCS1


                                        ;SAVE REGISTERS FOR COMPARISON AFTER NORMAL READ
021116  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
021122  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
021124  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
021126  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

021130  004737  030140                  JSP     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV


021134  013777  004472  161016          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


021142  013746  002334                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
021146  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
021152  012677  161014                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;70 IN RHCS1 FOR READ DATA
                                                                ;WITH INTERRUPT ENABLED


021156  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
021160  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
021162  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
021164  001614                          908.                    ;ALLOW 9000 MICRO SECONDS
021166  001507                          839.                    ;RDY MUST SET BETWEEN
                                                                ;690 AND 17470 MICRO SECONDS


                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
021170  004037  027762                  JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
021174  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
021176  004420                          REINTO+<256.*2>                 ;DATA
021200  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
021204  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
021206  000000                          0                       ;DATA
021210  004037  027762                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
021214  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
021216  000001                          1                       ;DATA
021220  017746  160744                  MOV     @RHCS2,-(SP)    ;GET RHCS2
021224  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
021230  042737  000300  004502          BIC     #IP!OR,@#SAVERE+4;CLEAR SAVED IR OR
021236  052637  004502                  BIS     (SP)+,@#SAVERE+4;SET OR IP AS REQUIRED
```

```
                                        ;COMPARE REGISTERS BEFORE READ WITH AFTER
                                        ;
021242  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
021246  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
021250  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
021252  000022                          18.                     ;18. REGISTERS TO BE COMPARED
021254  021260                          3$                      ;RETURN TO 3$ ON ERROR
021256  021264                          4$                      ;RETURN TO 4$ ON NO ERROR
                                        ;
021260  104033                  3$:     ERROR   33              ;READ DATA AFTER AN
021262  000207                          RTS     PC              ;ATTEMPTED WRITE WITH WRONG
                                                                ;IMPROPER REGISTER CHANGE
                                                                ;FORMAT CAUSED
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER READ
                                        ;
                                        ;COMPARE DATA READ AFTER ATTEMPTED WRITE WITH
                                        ;WRONG FORMAT BIT
021264                          4$:
                                        ;
021264  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
021270  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
021272  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
021274  000400                          256.                    ;256. WORDS TO BE COMPARED
021276  021302                          5$                      ;RETURN TO 5$ ON ERROR
021300  021306                          6$                      ;RETURN TO 6$ ON NO ERROR
                                        ;
021302  104034                  5$:     ERROR   34              ;DATA READ AFTER AN ATTEMPT
021304  000207                          RTS     PC              ;TO WRITE WITH WRONG FORMAT
                                                                ;WAS INCORRECT
                                        ;
021306                          6$:
                                        ;
                                        ;*****************************************************************
                                        ;*TEST 25       FORMAT ERROR RHER1 BIT #4
                                        ;
                                        ;*      AN ATTEMPT IS MADE TO READ DATA WITH WRONG
                                        ;*      FORMAT BIT
                                        ;*
                                        ;*      FORMAT ERROR BIT #4 IN RHER1 SHOULD SET
                                        ;*      NO DATA SHOULD BE READ
                                        ;
                                        ;*****************************************************************
021306  000004                  TST25:  SCOPE
021310  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021314  012737  000025  004470          MOV     #25,@#TSTNM     ;SAVE TEST NUMBER
                                        ;
021322  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
```

```
5490
5491                                        ;FILL WRITE FROM BUFFER WITH 107070
5492  021326  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
5493  021332  002354                        WRFROM                  ;STARTING FROM WRFROM
5494  021334  000400                        256.                    ;256. WORDS
5495  021336  107070                        107070                  ;FILL WITH 107070
5496
5497
5498                                        ;FILL READ INTO BUFFER WITH 107070
5499  021340  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
5500  021344  003420                        REINTO                  ;STARTING FROM REINTO
5501  021346  000400                        256.                    ;256. WORDS
5502  021350  107070                        107070                  ;FILL WITH 107070
5503
5504
5505                                        ;FILL COMMAND TO READ WITH WRONG FORMAT
5506
5507  021352  004037  032024                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5508  021356  000000                        0                       ;CYLINDER 0
5509  021360     000                .BYTE   0                       ;SECTOR 0
5510  021361     000                .BYTE   0                       ;TRACK 0
5511  021362  177400                        -256.                   ;WORD COUNT = 256.
5512  021364  003420                        REINTO                  ;BUS ADDRESS
5513                                                                ;STARTING ADDRESS OF DATA
5514                                                                ;BUFFER = REINTO
5515  021366  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5516  021370  004000                        ECI                     ;18 BITS PER WORD FORMAT
5517                                                                ;INHIBIT ECC CORRECTION
5518                                                                ;DO NOT INHIBIT HEADER COMPARE
5519  021372  002334                        READAT                  ;GET READY TO DO A READAT
5520                                              ;READ DATA WITH 70 IN RHCS1
5521
5522
5523                                        ;SAVE REGISTERS FOR COMPARAISON AFTER READ
5524  021374  004037  030220                JSR     R0,@#SAVER      ;SAVE REGISTERS
5525  021400  002164                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5526  021402  004476                        SAVERE                  ;STARTING ADDRES OF WHERE
5527                                                                ;THE REGISTERS ARE SAVED
5528  021404  000022                        18.                     ;NUMBER OF REGISTERS
5529                                                                ;SAVED = 18.
5530
5531  021406  004737  030140                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5532
5533
5534  021412  013777  004472  160540        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5535                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
5536                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5537                                                                ;'TIME' WILL ONLY SAVE
5538                                                                ;CURRENT CYLINDER ADDRESS
5539                                                                ;AND LOOK AHEAD REGISTERS
5540
5541
5542  021420  013746  002334                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
5543  021424  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5544                                                                ;ENABLE INTERRUPT
5545  021430  012677  160536                MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;7# IN RHCS1 FOR READ DATA
                                                       ;WITH INTERRUPT ENABLED


021434  104420                        WAT             ;WAIT FOR RDY BIT TO SET
021436  002172                        RHCS1           ;WAIT FOR RHCS1 REGISTER
021440  000200                        RDY             ;WAIT FOR RDY BIT IN RHCS1 REGISTER
021442  001522                        #50.            ;ALLOW #500 MICRO SECONDS
021444  001510                        #40.            ;RDY MUST SET BETWEEN
                                                       ;100 AND 16900 MICRO SECONDS


                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUE


021446  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021452  002172                        RHCS1           ;CHANGE RHCS1 REGISTER

021454  000002                        2               ;2 BIT/BITS TO BE CHANGED
021456  000001                        1               ;NEW VALUE OF SC IS 1
021460  100000                        SC              ;CHANGE SC BIT
021462  000001                        1               ;NEW VALUE OF TRE IS 1
021464  040000                        TRE             ;CHANGE TRE BIT

021466  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021472  002214                        RHDS1           ;CHANGE RHDS1 REGISTER

021474  000002                        2               ;2 BIT/BITS TO BE CHANGED
021476  000001                        1               ;NEW VALUE OF ATA IS 1
021500  100000                        ATA             ;CHANGE ATA BIT
021502  000001                        1               ;NEW VALUE OF ERR IS 1
021504  040000                        ERR             ;CHANGE ERR BIT
021506  004037  027762                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
021512  002176                        RHDST           ;SAVED REGISTER TO CHANGE
021514  000001                        1               ;DATA

021516  004037  030732                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021522  002174                        RHER1           ;CHANGE RHER1 REGISTER

021524  000001                        1               ;1 BIT/BITS TO BE CHANGED
021526  000001                        1               ;NEW VALUE OF FER IS 1
021530  000020                        FER             ;CHANGE FER BIT

021532  053737  004622  004522        BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
                                                                        ;FOR WORKING DRIVE IN
                                                                        ;SAVED RHAS LOACTION
021540  017746  160424                MOV     @RHCS2,-(SP)    ;GET RHCS2
021544  042716  177477                BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
021550  042737  000300  004502        BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
021556  052637  004502                BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED


                                ;COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT

021562  004037  031040                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
```

```
5602  021566  004476                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5603  021570  002242                        WC                      ;TEST DATA STARTING FROM 'RHWC'
5604  021572  000022                        18.                     ;18. REGISTERS TO BE COMPARED
5605  021574  021600                        1$                      ;RETURN TO 1$ ON ERROR
5606  021576  021634                        2$                      ;RETURN TO 2$ ON NO ERROR
5607
5608  021600  104057              1$:       ERROR   57              ;ATTEMPTING TO READ DATA
5609  021602  000207                        RTS     PC              ;WITH WRONG FORMAT BIT CAUSED
5610                                                                ;IMPROPER REGISTER CHANGE
5611                                                                ;GOOD DATA GIVES WHAT SHOULD BE
5612                                                                ;THERE
5613                                                                ;RECEIVED DATA GIVES WHAT WAS THERE
5614                                                                ;AFTER READ DATA
5615
5616                                        ;COMPARE READ INTO BUFFER TO CHECK THAT NOTHING WAS READ
5617  021604                        2$:
5618
5619  021604  004037  032070                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5620  021610  002354                        WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
5621  021612  003420                        REINTO                  ;TEST DATA STARTS FROM REINTO
5622  021614  000400                        256.                    ;256. WORDS TO BE COMPARED
5623  021616  021622                        3$                      ;RETURN TO 3$ ON ERROR
5624  021620  021626                        4$                      ;RETURN TO 4$ ON NO ERROR
5625
5626
5627  021622  104034              3$:       ERROR   34              ;ATTEMPT TO READ
5628  021624  000207                        RTS     PC              ;WITH WRONG FORMAT BIT
5629                                                                ;CHANGED READ INTO BUFFER
5630                                                                ;GOOD DATA GIVES WHAT SHOULD
5631                                                                ;BE THERE
5632                                                                ;BAD DATA GIVES WHAT WAS
5633                                                                ;THERE AFTER READ DATA
5634
5635  021626                        4$:
5636
5637
5638                                        ;*********************************************************************
5639                                        ;*TEST 26       REGISTER MODIFICATION REFUSED BIT #2 RHER1
5640
5641                                        ;*      CYLINDER1 TRACK 0, SECTOR 0 WILL BE WRITTEN WITH
5642                                        ;*      200 WORDS OF 2000 BY A WRITE HEADER AND DATA COMMAND
5643                                        ;*
5644                                        ;*      THE HEADS WILL BE BROUGHT TO CYLINDER 0 BY A SEEK
5645                                        ;*
5646                                        ;*      A READ DATA COMMAND WILL BE GIVEN TO CYLINDER 1 TRACK 0
5647                                        ;*      SECTOR 0 150. WORDS. THIS WILL TAKE AT
5648                                        ;*      LEAST 7 MILI SECONDS.  IMMEDIATELY AFTER GO AT
5649                                        ;*      IMPLIED SEEK TIME, WRITE INTO A REGISTER WILL BE ATTMEPTED
5650                                        ;*      THEN READY WILL BE WAITED ON TO COMPLETE THE READ DATA
5651                                        ;*      THEN ALL REGISTERS WILL BE COMPARED AND THE DATA READ
5652                                        ;*      SHOULD BE GOOD
5653                                        ;*      THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST, RHER2
5654                                        ;*      RHOF, RHCA, RHER3
5655
5656                                        ;*********************************************************************
5657  021626  000004                TST26:  SCOPE
```

```
5658  021630  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5659  021634  012737  000026  004470          MOV     #26,@#TSTNM     ;SAVE TEST NUMBER
5660
5661  021642  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5662                                                                  ;R3-RHDS1, R4-RHER1
5663                                                                  ;GIVE RH-11 INITIALIZE
5664                                                                  ;SETUP UNIT NUBER
5665  021646  012737  002172  004626          MOV     #RHCS1,@#TMP0   ;FIRST REGISTER TO BE TESTED
5666  021654  012737  000007  004634          MOV     #7,@#TMP5               ;NUMBER OF REGISTERS TO BE TESTED
5667                                          ;PREPARE TO WRITE HEADER AND DATA CYLINDER 1, TRACK 0, SECTOR 0
5668                                          ;FILL WRITE FROM BUFFER WITH HEADER
5669  021662                          ST22:
5670
5671  021662  004737  030060                  JSR     PC,@#CLDISK     ;SET P1-RHCS1, R2-RHCS2
5672                                                                  ;R3-RHDS1, R4-RHER1
5673                                                                  ;GIVE RH-11 INITIALIZE
5674                                                                  ;SETUP UNIT NUBER
5675
5676  021666  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
5677  021672  002354                          WRFROM                  ;LOCATION WHERE SAVED
5678  021674  000004                          4                       ;NUMBER OF WORDS SAVED
5679  021676  010001                          10001                   ;FIRST DATA WORD
5680  021700  000000                          0                       ;SECOND DATA WORD
5681  021702  000000                          0                       ;THIRD DATA WORD
5682  021704  000000                          0                       ;FOURTH DATA WORD
5683
5684                                          ;FILL WRITE FROM BUFFER WITH DATA
5685  021706  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
5686  021712  002364                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
5687  021714  000400                          256.                    ;256. WORDS
5688  021716  002000                          2000                    ;FILL WITH 2000
5689
5690                                          ;FILL COMMAND
5691
5692  021720  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5693  021724  000001                          1                       ;CYLINDER 1
5694  021726     000                  .BYTE   0                       ;SECTOR 0
5695  021727     000                  .BYTE   0                       ;TRACK 0
5696  021730  177464                          -200.-4         ;WORD COUNT (DATA)=200.+
5697                                                                  ;4 HEADER WORDS
5698  021732  002354                          WRFROM                  ;BUS ADDRESS
5699                                                                  ;STARTING ADDRESS OF DATA
5700                                                                  ;BUFFER = WRFROM
5701  021734  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5702  021736  010000                          FMT22                   ;16 BITS PER WORD FORMAT
5703                                                                  ;DO NOT INHIBIT ECC CORRECTION
5704                                                                  ;DO NOT INHIBIT HEADER COMPARE
5705  021740  002332                          WRIFOR                  ;GET READY TO DO A WRIFOR
5706                                                          ;WRITE HEADER AND DATA WITH 62 IN RHCS1
5707
5708
5709  021742  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5710
5711
5712  021746  013777  004472  160204          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5713                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
```

```
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDPESS
                                                        ;AND LOOK AHEAD REGISTERS

021754  013746  002332              MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
021760  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
021764  012677  160202              MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                        ;WITH INTERRUPT ENABLED
                                    ;TIME IS NOT IMPORTANT

021770  104420                      WAT                     ;WAIT FOR RDY BIT TO SET
021772  002172                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
021774  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
021776  004704                      2500.                   ;ALLOW 25000 MICRO SECONDS
022000  004704                      2500.                   ;RDY MUST SET BETWEEN
                                                        ;00 AND 50000 MICRO SECONDS

                                    ;NOW BRING THE HEADS TO CYLINDER 0

022002  004737  030140              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV


022006  004037  030030              JSR     R0,@#SEEKCY     ;SEEK FOR
022012  000000                      0                       ;CYLINDER 0

022014  013777  004472  160136      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS


022022  013746  002340              MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
022026  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
022032  012677  160134              MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;4 IN RHCS1 FOR SEEK
                                                        ;WITH INTERRUPT ENABLED


022036  104420                      WAT                     ;WAIT FOR DRY BIT TO SET
022040  002214                      RHDS1                   ;WAIT FOR RHDS1 REGISTER
022042  000200                      DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
022044  002776                      1534.                   ;ALLOW 15340 MICRO SECONDS
022046  001502                      834.                    ;DRY MUST SET BETWEEN
                                                        ;7000 AND 23680 MICRO SECONDS

                                    ;PREPARE FOR A READ DATA

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

```
5770  022050  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 150. WORDS, FROM WRFROM
5771  022054  002354                          WRFROM                  ;STARTING FROM WRFROM
5772  022056  000226                          150.                    ;150. WORDS
5773  022060  002000                          2000                    ;FILL WITH 2000
5774
5775  022062  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 106. WORDS, FROM WRFROM+<150.*2>
5776  022066  003030                          WRFROM+<150.*2>                 ;STARTING FROM WRFROM+<150.*2>
5777  022070  000152                          106.                    ;106. WORDS
5778  022072  000077                          77                      ;FILL WITH 77
5779
5780
5781                                          ;FILL READ INTO BUFFER WITH DATA OTHER THAN WHAT IS EXPECTED
5782  022074  004037  032730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
5783  022100  003420                          REINTO                  ;STARTING FROM REINTO
5784  022102  000400                          256.                    ;256. WORDS
5785  022104  000077                          77                      ;FILL WITH 77
5786
5787                                          ;FILL READ DATA COMMAND
5788
5789  022106  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5790  022112  000001                          1                       ;CYLINDER 1
5791  022114     000                  .BYTE   0                       ;SECTOR 0
5792  022115     000                  .BYTE   0                       ;TRACK 0
5793  022116  177552                          -150.                   ;WORD COUNT = 150.
5794  022120  003420                          REINTO                  ;BUS ADDRESS
5795                                                                  ;STARTING ADDRESS OF DATA
5796                                                                  ;BUFFER = REINTO
5797  022122  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5798  022124  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
5799                                                                  ;INHIBIT ECC CORRECTION
5800                                                                  ;DO NOT INHIBIT HEADER COMPARE
5801  022126  002334                          READAT                  ;GET READY TO DO A READAT
5802                                                  ;READ DATA WITH 70 IN RHCS1
5803
5804
5805                                          ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE
5806                                          ;INTO A REGISTER WHILE THE READ IS GOING ON
5807  022130  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
5808  022134  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5809  022136  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
5810                                                                  ;THE REGISTERS ARE SAVED
5811  022140  000022                          18.                     ;NUMBER OF REGISTERS
5812                                                                  ;SAVED = 18.
5813
5814  022142  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5815
5816
5817  022146  013777  004472  160004          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5818                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5819                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5820                                                                  ;'TIME' WILL ONLY SAVE
5821                                                                  ;CURRENT CYLINDER ADDRESS
5822                                                                  ;AND LOOK AHEAD REGISTERS
5823
5824
5825  022154  013746  002334                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
```

```
5826  022160  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5827                                                                  ;ENABLE INTERRUPT
5828  022164  012677  160002                  MOV     (SP)+,@RHCS1    ;GO WITH
5829                                                                   ;70 IN RHCS1 FOR READ DATA
5830                                                                  ;WITH INTERRUPT ENABLED
5831
5832  022170  013700  004626                  MOV     @#TMP0,R0               ;SET UP R0 FOR WRITE
5833  022174  012730  002006                  MOV     #BIT1!BIT2!BIT10,@(R0)+ ;ATTEMPT TO WRITE INTO
5834                                                                  ;REGISTERS DURING IMPLIED SEEK
5835  022200  010037  004626                  MOV     R0,@#TMP0               ;SAVE OFF  R0
5836
5837                                          ;NOW RMR IS SET BUT THE COMPLETION OF READ MUST BE
5838                                          ;WAITED ON
5839
5840
5841  022204  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
5842  022206  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
5843  022210  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5844  022212  002237                          1183.                   ;ALLOW 11830 MICRO SECONDS
5845  022214  002237                          1183.                   ;RDY MUST SET BETWEEN
5846                                                                  ;00 AND 23660 MICRO SECONDS
5847                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
5848
5849  022216  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5850  022222  002226                          RHCC                    ;CHANGE RHCC REGISTER
5851
5852  022224  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5853  022226  000001                          1                       ;NEW VALUE OF BIT0 IS 1
5854  022230  000001                          BIT0                    ;CHANGE BIT0 BIT
5855
5856  022232  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5857  022236  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
5858
5859  022240  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5860  022242  000001                          1                       ;NEW VALUE OF SC IS 1
5861  022244  100000                          SC                      ;CHANGE SC BIT
5862  022246  004037  027762                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5863  022252  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
5864  022254  000001                          1                       ;DATA
5865
5866  022256  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5867  022262  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
5868
5869  022264  000002                          2                       ;2 BIT/BITS TO BE CHANGED
5870  022266  000001                          1                       ;NEW VALUE OF ATA IS 1
5871  022270  100000                          ATA                     ;CHANGE ATA BIT
5872  022272  000001                          1                       ;NEW VALUE OF ERR IS 1
5873  022274  040000                          ERR                     ;CHANGE ERR BIT
5874
5875  022276  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
5876                                                                  ;FOR WORKING DRIVE IN
5877                                                                  ;SAVED RHAS LOACTION
5878
5879  022304  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5880  022310  002174                          RHER1                   ;CHANGE RHER1 REGISTER
5881
```

```
5882  022312  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5883  022314  000001                          1                       ;NEW VALUE OF RMR IS 1
5884  022316  000004                          RMR                     ;CHANGE RMR BIT
5885  022320  004037  027762                  JSR     R0,@#FILLRE     ;MOV REINTO+<150.*2> INTO SAVED RHBA
5886  022324  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
5887  022326  004074                          REINTO+<150.*2>                 ;DATA
5888  022330  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
5889  022334  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
5890  022336  000000                          0                       ;DATA
5891                                          ;COMPARE REGISTERS BEFORE READ DATA WITH REGISTERS
5892                                          ;AFTER READ AND ATTEMPTED MODIFICATION OF REGISTER
5893
5894  022340  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5895                                                                  ;PRESENT VALUE
5896  022344  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5897  022346  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5898  022350  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5899  022352  022356                          2$                      ;RETURN TO 2$ ON ERROR
5900  022354  022376                          3$                      ;RETURN TO 3$ ON NO ERROR
5901
5902  022356                          2$:
5903  022356  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5904  022360  013700  004626                  MOV     @#TMP0,R0       ;GET REGISTER BEEING MODIFYED + 2 POINTER
5905  022364  014037  001122                  MOV     -(R0),@#BDADR   ;GET ADDRESS OF REGISTER BEING MODIFYED
5906  022370  104060                          ERROR   60              ;ATTEMPTING TO MODIFY REGISTER
5907  022372  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5908  022374  000207                          RTS     PC              ;DURING A READ COMMAND CAUSED
5909                                                                  ;IMPROPER REGISTER CHANGE
5910                                                                  ;GOOD DATA GIVES WHAT SHOULD
5911                                                                  ;BE THERE
5912                                                                  ;RECEIVED DATA GIVES WHAT WAS
5913                                                                  ;THERE AFTER READ
5914                                  ;COMPARE DATA READ
5915  022376                          3$:
5916
5917  022376  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5918  022402  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
5919  022404  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
5920  022406  022412                          4$                      ;4$ WORDS TO BE COMPARED
5921  022410  022416                          ST23                    ;RETURN TO ST23 ON ERROR
5922                                                                  ;RETURN TO  ON NO ERROR
5923
5924
5925  022412  104034                  4$:     ERROR   34              ;DATA READ WITH AN ATTEMPTED
5926  022414  000207                          RTS     PC              ;MODIFICATION OF REGISTER
5927                                                                  ;DURING READ CAUSED ERROR
5928  022416  005337  004634          ST23:   DEC     @#TMP5          ;COUNT DOWN
5929  022422  001002                          BNE     1$              ;BRANCH IF 7 NOT DONE
5930  022424  000167  000004                  JMP     TST27   ;JUMP TO NEXT TEST
5931  022430  000137  021662          1$:     JMP     @#ST22          ;JUMP TO BEGINING OF TEST
5932
5933                                  ;*********************************************************************
5934                                  ;*TEST 27       REGISTER MODIFICATION REFUSED RMR-BIT#2 IN RHER1
5935
5936                                  ;*      A WRITE HEADER AND DATA COMMAND WILL BE GIVEN TO
5937                                  ;*      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
                                ;*****************************************************************
022434  000004                 TST27:  SCOPE
022436  012706  001000                 MOV     #STACK,SP       ;RESET STACK
022442  012737  000027  004470         MOV     #27,@#TSTNM     ;SAVE TEST NUMBER
022450  004737  030060                 JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
022454  012737  002172  004626         MOV     #RHCS1,@#TMP0   ;FILL REGISTER TO BE MODIFIED
022462  012737  000007  004634         MOV     #7,@#TMP5               ;NUMBER OF REGISTERS TO BE TESTED
                                ;PREPARE TO WRITE HEADER AND DATA
022470                          ST24:
022470  004737  030060                 JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                ;FILL WRITE FROM BUFFER WITH HEADER
022474  004037  027704                 JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
022500  002354                         WRFROM                  ;LOCATION WHERE SAVED
022502  000004                         4                       ;NUMBER OF WORDS SAVED
022504  010001                         10001                   ;FIRST DATA WORD
022506  000000                         0                       ;SECOND DATA WORD
022510  000000                         0                       ;THIRD DATA WORD
022512  000000                         0                       ;FOURTH DATA WORD
                                ;FILL WRITE FROM BUFFER WITH DATA
022514  004037  027730                 JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
022520  002364                         WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
022522  000400                         256.                    ;256. WORDS
022524  070707                         070707                  ;FILL WITH 070707
```

```
5994
5995                                        ;FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
5996
5997  022526  004037  027704                JSR     P0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
5998  022532  003364                        WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
5999  022534  000004                        4                       ;NUMBER OF WORDS SAVED
6000  022536  010001                        10001                   ;FIRST DATA WORD
6001  022540  000001                        1                       ;SECOND DATA WORD
6002  022542  000000                        0                       ;THIRD DATA WORD
6003  022544  000000                        0                       ;FOURTH DATA WORD
6004
6005                                        ;FILL WRITE FROM BUFFER WITH WITH NEXT SECTOR DATA
6006  022546  004037  027730                JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<268.*2>
6007  022552  003404                        WRFROM+<268.*2>                 ;STARTING FROM WRFROM+<268.*2>
6008  022554  000004                        4                       ;4 WORDS
6009  022556  070707                        70707                   ;FILL WITH 70707
6010
6011
6012                                        ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
6013
6014  022560  004037  032024                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6015  022564  000001                        1                       ;CYLINDER 1
6016  022566     000                .BYTE   0                       ;SECTOR 0
6017  022567     000                .BYTE   0                       ;TRACK 0
6018  022570  177364                        -264.-4         ;WORD COUNT (DATA)=264.+
6019                                                        ;4 HEADER WORDS
6020  022572  002354                        WRFROM                  ;BUS ADDRESS
6021                                                        ;STARTING ADDRESS OF DATA
6022                                                        ;BUFFER = WRFROM
6023  022574  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6024  022576  010000                        FMT22                   ;16 BITS PER WORD FORMAT
6025                                                        ;DO NOT INHIBIT ECC CORRECTION
6026                                                        ;DO NOT INHIBIT HEADER COMPARE
6027  022600  002332                        WRIFOR                  ;GET READY TO DO A WRIFOR
6028                                              ;WRITE HEADER AND DATA WITH 62 IN RHCS1
6029
6030
6031  022602  004737  030140                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6032
6033
6034  022606  013777  004472  157344        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6035                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
6036                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6037                                                        ;'TIME' WILL ONLY SAVE
6038                                                        ;CURRENT CYLINDER ADDRESS
6039                                                        ;AND LOOK AHEAD REGISTERS
6040
6041
6042  022614  013746  002332                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
6043  022620  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
6044                                                        ;ENABLE INTERRUPT
6045  022624  012677  157342                MOV     (SP)+,@RHCS1    ;GO WITH
6046                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
6047                                                        ;WITH INTERRUPT ENABLED
6048                                        ;ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6049
```

```
6050  022630  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
6051  022632  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6052  022634  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6053  022636  001725                          981.                    ;ALLOW 9810 MICRO SECONDS
6054  022640  001502                          834.                    ;RDY MUST SET BETWEEN
6055                                                                  ;1470 AND 18150 MICRO SECONDS
6056
6057                                          ;NOW PREPARE FOR THE WRITE DATA COMMAND
6058
6059                                          ;FILL WRITE FROM BUFFER WITH 256 OF 2000 AND 4 OF 2001
6060  022642  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
6061  022646  002354                          WRFROM                  ;STARTING FROM WRFROM
6062  022650  000400                          256.                    ;256. WORDS
6063  022652  002000                          2000                    ;FILL WITH 2000
6064
6065  022654  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
6066  022660  003354                          WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
6067  022662  000004                          4                       ;4 WORDS
6068  022664  002001                          2001                    ;FILL WITH 2001
6069
6070
6071                                          ;FILL WRITE DATA COMMAND
6072
6073  022666  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6074  022672  000001                          1                       ;CYLINDER 1
6075  022674     000                  .BYTE   0                       ;SECTOR 0
6076  022675     000                  .BYTE   0                       ;TRACK 0
6077  022676  177400                          -256.                   ;WORD COUNT = 256.
6078  022700  002354                          WRFROM                  ;BUS ADDRESS
6079                                                                  ;STARTING ADDRESS OF DATA
6080                                                                  ;BUFFER = WRFROM
6081  022702  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6082  022704  010000                          FMT22                   ;16 BITS PER WORD FORMAT
6083                                                                  ;DO NOT INHIBIT ECC CORRECTION
6084                                                                  ;DO NOT INHIBIT HEADER COMPARE
6085  022706  002330                          WRIDAT                  ;GET READY TO DO A WRIDAT
6086                                                        ;WRITE DATA WITH 60 IN RHCS1
6087
6088
6089                                          ;SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED
6090                                          ;REGISTER MODIFICATION DURING A WRITE DATA
6091  022710  004037  030220                  JSR     P0,@#SAVER      ;SAVE REGISTERS
6092  022714  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6093  022716  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
6094                                                                  ;THE REGISTERS ARE SAVED
6095  022720  000022                          18.                     ;NUMBER OF REGISTERS
6096                                                                  ;SAVED = 18.
6097
6098  022722  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6099
6100
6101  022726  013777  004472  157224          MOV     @#RP4VEC,@RPVEC ;SET RP#4 VECTOR ADDRESS
6102                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6103                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6104                                                                  ;'TIME' WILL ONLY SAVE
6105                                                                  ;CURRENT CYLINDER ADDRESS
```

```
                                                        ;AND LOOK AHEAD REGISTERS


022734  013746  002330                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
022740  052716  000101                BIS     #GO!IE,(SP)     ;GET PEADY TO SET GO AND
                                                        ;ENABLE INTERRUPT
022744  012677  157222                MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;60 IN RHCS1 FOR WRITE DATA
                                                        ;WITH INTERRUPT ENABLED

022750  013700  004626                MOV     @#TMP0,R0               ;SET R0 TO REG ADDRESS
022754  012730  002002                MOV     #BIT1!BIT10,@(R0)+ ;ATTEMPT TO WRITE INTO A REGISTER
                                                        ;DURING WRITE DATA
022760  010037  004626                MOV     R0,@#TMP0               ;SAVE OFF NEW REG ADDRESS

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

022764  104420                      WAT                     ;WAIT FOR RDY BIT TO SET
022766  002172                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
022770  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
022772  001725                      981.                    ;ALLOW 9810 MICRO SECONDS
022774  001502                      834.                    ;RDY MUST SET BETWEEN
                                                        ;1470 AND 10150 MICRO SECONDS

                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUE

022776  004037  030732              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
023002  002226                      RHCC                    ;CHANGE RHCC REGISTER

023004  000001                      1                       ;1 BIT/BITS TO BE CHANGED
023006  000001                      1                       ;NEW VALUE OF BIT0 IS 1
023010  000001                      BIT0                    ;CHANGE BIT0 BIT

023012  004037  030732              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
023016  002172                      RHCS1                   ;CHANGE RHCS1 REGISTER

023020  000001                      1                       ;1 BIT/BITS TO BE CHANGED
023022  000001                      1                       ;NEW VALUE OF SC IS 1
023024  100000                      SC                      ;CHANGE SC BIT

023026  004037  030732              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
023032  002214                      RHDS1                   ;CHANGE RHDS1 REGISTER

023034  000002                      2                       ;2 BIT/BITS TO BE CHANGED
023036  000001                      1                       ;NEW VALUE OF ATA IS 1
023040  100000                      ATA                     ;CHANGE ATA BIT
023042  000001                      1                       ;NEW VALUE OF ERR IS 1
023044  040000                      ERR                     ;CHANGE ERR BIT

023046  053737  004622  004522      BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
                                                        ;FOR WORKING DRIVE IN
                                                        ;SAVED RHAS LOACTION

023054  004037  030732              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
023060  002174                      RHER1                   ;CHANGE RHER1 REGISTER
```

```
6162  023062  000001                          1                       ;1 BIT/BITS TO BE CHANGED
6163  023064  000001                          1                       ;NEW VALUE OF RMR IS 1
6164  023066  000004                          RMR                     ;CHANGE RMR BIT
6165  023070  004037  027762                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6166  023074  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
6167  023076  000001                          1                       ;DATA
6168  023100  004037  027762                  JSR     R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
6169  023104  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
6170  023106  003354                          WRFROM+<256.*2>                 ;DATA
6171  023110  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6172  023114  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
6173  023116  000000                          0                       ;DATA
6174                                          ;COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6175                                          ;AFTER WRITE AND ATTEMPTED MODIFICATION OF REGISTER
6176
6177  023120  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6178                                                                  ;PRESENT VALUE
6179  023124  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6180  023126  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6181  023130  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6182  023132  023136                          2$                      ;RETURN TO 2$ ON ERROR
6183  023134  023156                          3$                      ;RETURN TO 3$ ON NO ERROR
6184
6185  023136                          2$:
6186  023136  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6187  023140  013700  004626                  MOV     @#TMP0,R0       ;GET REGISTER BEEING MODIFYED + 2 POINTER
6188  023144  014037  001122                  MOV     -(R0),@#$BDADR  ;GET ADDRESS OF REGISTER BEING MODIFYED
6189  023150  104060                          ERROR   60              ;ATTEMPTING TO MODIFY REGISTER
6190  023152  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
6191  023154  000207                          RTS     PC              ;DURING A WRITE COMMAND CAUSED
6192                                                                  ;IMPROPER REGISTER GIVES WHAT SHOULD
6193                                                                  ;GOOD DATA GIVES WHAT SHOULD
6194                                                                  ;BE THERE
6195                                                                  ;RECEIVED DATA GIVES WHAT WAS
6196                                                                  ;THERE AFTER READ
6197                                          ;CLEAR ALL ERROR FLAGS
6198  023156                          3$:
6199
6200  023156  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6201                                                                  ;R3-RHDS1, R4-RHER1
6202                                                                  ;GIVE RH-11 INITIALIZE
6203                                                                  ;SETUP UNIT NUBER
6204                                          ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
6205  023162  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
6206  023166  002354                          WRFROM                  ;STARTING FROM WRFROM
6207  023170  000400                          256.                    ;256. WORDS
6208  023172  002000                          2000                    ;FILL WITH 2000
6209
6210  023174  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
6211  023200  003354                          WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
6212  023202  000004                          4                       ;4 WORDS
6213  023204  002001                          2001                    ;FILL WITH 2001
6214
6215                                          ;NOW THE READ DATA COMMAND WILL BE FILLED
6216
6217  023206  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
```

```
6219  023212  000001                          1                       ;CYLINDER 1
6219  023214     000                .BYTE     0                       ;SECTOR 0
6220  023215     000                .BYTE     0                       ;TRACK 0
6221  023216  177374                          -260.                   ;WORD COUNT = 260.
6222  023220  003420                          REINTO                  ;BUS ADDRESS
6223                                                                  ;STARTING ADDRESS OF DATA
6224                                                                  ;BUFFER = REINTO
6225  023222  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6226  023224  014000                          FCI!FMT22                       ;16 BITS PER WORD FORMAT
6227                                                                  ;INHIBIT ECC CORRECTION
6228                                                                  ;DO NOT INHIBIT HEADER COMPARE
6229  023226  002334                          READAT                  ;GET READY TO DO A READAT
6230                                                        ;READ DATA WITH 70 IN RHCS1
6231
6232
6233                                          ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6234  023230  004037  030220                  JSR       P0,@#SAVER    ;SAVE REGISTERS
6235  023234  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6236  023236  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
6237                                                                  ;THE REGISTERS ARE SAVED
6238  023240  000022                          18.                     ;NUMBER OF REGISTERS
6239                                                                  ;SAVED = 18.
6240
6241  023242  004737  030140                  JSR       PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6242
6243
6244  023246  013777  004472  156704          MOV       @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6245                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6246                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6247                                                                  ;'TIME' WILL ONLY SAVE
6248                                                                  ;CURRENT CYLINDER ADDRESS
6249                                                                  ;AND LOOK AHEAD REGISTERS
6250
6251
6252  023254  013746  002334                  MOV       @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
6253  023260  052716  000101                  BIS       #GO!IE,(SP)   ;GET READY TO SET GO AND
6254                                                                  ;ENABLE INTERRUPT
6255  023264  012677  156702                  MOV       (SP)+,@RHCS1  ;GO WITH
6256                                                                   ;70 IN RHCS1 FOR READ DATA
6257                                                                  ;WITH INTERRUPT ENABLED
6258  023270  011100                          MOV       @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
6259  023272  011305                          MOV       @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
6260
6261
6262  023274  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
6263  023276  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6264  023300  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6265  023302  001725                          981.                    ;ALLOW 9810 MICRO SECONDS
6266  023304  001502                          834.                    ;RDY MUST SET BETWEEN
6267                                                                  ;1470 AND 18150 MICRO SECONDS
6268                                          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6269                                          ;R0 AND R5 IMMEDIATELY AFTER GO
6270  023306  013746  002334                  MOV       @#READAT,-(SP)  ;SAVE COMMAND
6271  023312  052716  004101                  BIS       #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
6272  023316  011637  001124                  MOV       (SP),@#GDDAT  ;SAVE FOR PRINTOUT
6273  023322  022600                          CMP       (SP)+,R0      ;DURING ABOVE OPERATION ONLY IE!DVA!GO
```

```
6274                                                    ;AND COMMAND SHOULD BE SET
6275  023324  001405                        BEQ     70$           ;BRANCH IF GOOD
6276  023326  010037  001126                MOV     R0,@#$BDDAT   ;BAD DATA
6277  023332  010137  004464                MOV     R1,@#REGADR   ;FAILING REGISTER RHCS1
6278  023336  104021                        ERROR   21            ;DURING ABOVE OPERATION ONLY
6279                                                    ;COMMAND AND IE!DVA!GO SHOULD BE SET
6280  023340  012746  010500          70$:  MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6281  023344  011637  001124                MOV     (SP),@#$GDDAT ;SAVE FOR PRINTOUT
6282  023350  022605                        CMP     (SP)+,R5      ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6283                                                    ;SHOULD BE SET
6284  023352  001405                        BEQ     72$           ;BRANCH IF GOOD
6285  023354  010537  001126                MOV     R5,@#$BDDAT   ;BAD DATA
6286  023360  010337  004464                MOV     R3,@#REGADR   ;FAILING REGISTER RHDS1
6287  023364  104063                        ERROR   63            ;DURING ABOVE OPERATION ONLY
6288                                                    ;MOL!DPR!VV SHOULD BE SET
6289  023366                          72$:
6290
6291                                                    ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
6292  023366  004037  027762                JSR     R0,@#FILLRE   ;MOV 0 INTO SAVED RHWC
6293  023372  002164                        RHWC                  ;SAVED REGISTER TO CHANGE
6294  023374  000000                        0                     ;DATA
6295
6296  023376  004037  027762                JSR     R0,@#FILLRE   ;MOV REINTO+<260.*2> INTO SAVED RHBA
6297  023402  002166                        RHBA                  ;SAVED REGISTER TO CHANGE
6298  023404  004430                        REINTO+<260.*2>               ;DATA
6299  023406  004037  027762                JSR     R0,@#FILLRE   ;MOV 2 INTO SAVED RHDST
6300  023412  002176                        RHDST                 ;SAVED REGISTER TO CHANGE
6301  023414  000002                        2                     ;DATA
6302
6303                                                    ;NOW COMPARE REGISTERS BEFORE READ DATA WITH
6304                                                    ;AFTER COMMAND
6305
6306  023416  004037  031040                JSR     R0,@#COMREG   ;COMPARE SAVED REGISTERS WITH
6307                                                    ;PRESENT VALUE
6308  023422  004476                        SAVERE                ;GOOD DATA SAVED IN 'SAVERE'
6309  023424  002242                        WC                    ;TEST DATA STARTING FROM 'RHWC'
6310  023426  000022                        18.                   ;18. REGISTERS TO BE COMPARED
6311  023430  023434                        4$                    ;RETURN TO 4$ ON ERROR
6312  023432  023440                        5$                    ;RETURN TO 5$ ON NO ERROR
6313
6314  023434  104033                  4$:   ERROR   33            ;READ DATA CAUSED IMPROPER REGISTER
6315  023436  000207                        RTS     PC            ;CHANGE
6316                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
6317                                                    ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
6318                                                    ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
6319                                                    ;WAS GOOD
6320  023440                          5$:
6321
6322  023440  004037  032070                JSR     R0,@#COMPAR   ;COMPARE TWO BLOCKS OF MEMORY
6323  023444  002354                        WRFROM                ;GOOD DATA      STARTS FROM WRFROM
6324  023446  003420                        REINTO                ;TEST DATA STARTS FROM REINTO
6325  023450  000400                        256.                  ;256. WORDS TO BE COMPARED
6326  023452  023456                        6$                    ;RETURN TO 6$ ON ERROR
6327  023454  023462                        ST2$                  ;RETURN TO ST2$ ON NO ERROR
6328
6329
```

```
6330  023456  104034                  6$:     ERROR   34              ;READ DATA ERROR AFTER A WRITE DATA
6331  023460  000207                          RTS     PC              ;WITH REGISTER MODIFICATION
6332                                                                  ;WITHIN THE WRITE DATA
6333                                          ;IF ALL 7 REGISTERS NOT COMPLETE THEN REPEAT
6334  023462  005337  004634          ST28:   DEC     @#TMP5          ;COUNT DOWN
6335  023466  001002                          BNE     1$              ;BRANCH IF 7 NOT DONE
6336  023470  000167  000004                  JMP     TST30   ;JUMP TO NEXT TEST
6337  023474  000167  176770          1$:     JMP     ST24            ;JUMP TO BEGINING OF TEST
6338
6339
6340                                  ;*****************************************************************
6341                                  ;*TEST 30       REGISTER MODIFICATION REFUSED BIT #2 RHER1
6342
6343                                  ;*      A READ DATA COMMAND IS GIVEN TO CYLINDER 0, SECTOR 0
6344                                  ;*      TRACK 0, IMMEDIATELY AFTER GO RHAS IS WRITTEN INTO
6345                                  ;*      WITH ALL ONES RMR BIT #2 IN RHER SHOULS NOT SET
6346
6347                                  ;*****************************************************************
6348  023500  000004                  TST30:  SCOPE
6349  023502  012706  001000                  MOV     #STACK,SP       ;RESET STACK
6350  023506  012737  000030  004470          MOV     #30,@#TSTNM     ;SAVE TEST NUMBER
6351
6352  023514  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6353                                                                  ;R3-RHDS1, R4-RHER1
6354                                                                  ;GIVE RH-11 INITIALIZE
6355                                                                  ;SETUP UNIT NUBER
6356
6357                                          ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
6358  023520  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
6359  023524  002354                          WRFROM                  ;STARTING FROM WRFROM
6360  023526  000400                          256.                    ;256. WORDS
6361  023530  000000                          0                       ;FILL WITH 0
6362
6363                                          ;FILL READ INTM BUFFER WITH ALL ONES
6364  023532  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
6365  023536  003420                          REINTO                  ;STARTING FROM REINTO
6366  023540  000400                          256.                    ;256. WORDS
6367  023542  177777                          -1                      ;FILL WITH -1
6368
6369                                          ;NOW THE READ DATA COMMAND WILL BE FILLED
6370
6371  023544  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6372  023550  000000                          0                       ;CYLINDER 0
6373  023552     000                  .BYTE   0                       ;SECTOR 0
6374  023553     000                  .BYTE   0                       ;TRACK 0
6375  023554  177400                          -256.                   ;WORD COUNT = 256.
6376  023556  003420                          REINTO                  ;BUS ADDRESS
6377                                                                  ;STARTING ADDRESS OF DATA
6378                                                                  ;BUFFER = REINTO
6379  023560  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6380  023562  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
6381                                                                  ;INHIBIT ECC CORRECTION
6382                                                                  ;DO NOT INHIBIT HEADER COMPARE
6383  023564  002334                          READAT                  ;GET READY TO DO A READAT
6384                                                          ;READ DATA WITH 70 IN RHCS1
6385
```

```
6386                                        ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6387  023566  004037  030220                JSR     R0,@#SAVER        ;SAVE REGISTERS
6388  023572  002164                        RHWC                      ;RHWC IS THE FIRST REGISTER SAVED
6389  023574  004476                        SAVERE                    ;STARTING ADDRES OF WHERE
6390                                                                  ;THE REGISTERS ARE SAVED
6391  023576  000022                        18.                       ;NUMBER OF REGISTERS
6392                                                                  ;SAVED = 18.
6393
6394  023600  004737  030140                JSR     PC,@#CHECKT       ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6395
6396
6397  023604  013777  004472  156346        MOV     @#RP4VEC,@PPVEC   ;SET RP04 VECTOR ADDRESS
6398                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6399                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6400                                                                  ;'TIME' WILL ONLY SAVE
6401                                                                  ;CURRENT CYLINDER ADDRESS
6402                                                                  ;AND LOOK AHEAD REGISTERS
6403
6404
6405
6406  023612  013746  002334                MOV     @#READAT,-(SP)    ;GET READY TO MOVE COMMAND
6407  023616  052716  000101                BIS     #GO!IE,(SP)       ;GET READY TO SET GO AND
6408                                                                  ;ENABLE INTERRUPT
6409  023622  012677  156344                MOV     (SP)+,@RHCS1      ;GO WITH
6410                                                                   ;70 IN RHCS1 FOR READ DATA
6411                                                                  ;WITH INTERRUPT ENABLED
6412  023626  011100                        MOV     @R1,R0            ;SAVE RHCS1 DURING ABOVE OPERATION
6413  023630  011305                        MOV     @R3,R5            ;SAVE RHDS1 DURING ABOVE OPERATION
6414
6415  023632  012777  177777  156350        MOV     #-1,@RHAS         ;WRITE INTO RHAS THIS SHOULD
6416                                                                  ;NOT SET RMP
6417
6418                                        ;TIME IS NOT IMPORTANT
6419
6420  023640  104420                        WAT                       ;WAIT FOR RDY BIT TO SET
6421  023642  002172                        RHCS1                     ;WAIT FOR RHCS1 REGISTER
6422  023644  000200                        RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6423  023646  003326                        1750.                     ;ALLOW 17500 MICRO SECONDS
6424  023650  000175                        125.                      ;RDY MUST SET BETWEEN
6425                                                                  ;16250 AND 18750 MICRO SECONDS
6426                                        ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6427                                        ;R0 AND R5 IMMEDIATELY AFTER GO
6428  023652  013746  002334                MOV     @#READAT,-(SP)    ;SAVE COMMAND
6429  023656  052716  004101                BIS     #IE!DVA!GO,(SP)   ;INCLUDE IE!DVA!GO
6430  023662  011637  001124                MOV     (SP),@#$GDDAT     ;SAVE FOR PRINTOUT
6431  023666  022600                        CMP     (SP)+,R0          ;DURING ABOVE OPERATION ONLY IE!DVA!GO
6432                                                                  ;AND COMMAND SHOULD BE SET
6433  023670  001405                        BEQ     64$               ;BRANCH IF GOOD
6434  023672  010037  001126                MOV     R0,@#$BDDAT       ;BAD DATA
6435  023676  010137  004464                MOV     R1,@#REGADR       ;FAILING REGISTER RHCS1
6436  023702  104021                        ERROR   21                ;DURING ABOVE OPERATION ONLY
6437                                                                  ;COMMAND AND IE!DVA!GO SHOULD BE SET
6438  023704  012746  010500          64$:  MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6439  023710  011637  001124                MOV     (SP),@#$GDDAT     ;SAVE FOR PRINTOUT
6440  023714  022605                        CMP     (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6441                                                                  ;SHOULD BE SET
```

```
6442  023716  001405                          BEQ     66$             ;BRANCH IF GOOD
6443  023720  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
6444  023724  010337  004464                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6445  023730  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
6446                                                                  ;MOL!DPR!VV SHOULD BE SET
6447  023732                          66$:
6448
6449                                          ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
6450  023732  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6451  023736  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
6452  023740  000000                          0                       ;DATA
6453  023742  004037  027762                  JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
6454  023746  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
6455  023750  004420                          REINTO+<256.*2>                 ;DATA
6456  023752  004037  027762                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6457  023756  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
6458  023760  000001                          1                       ;DATA
6459  023762  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
6460  023766  002226                          RHCC                    ;SAVED REGISTER TO CHANGE
6461  023770  000000                          0                       ;DATA
6462
6463                                          ;NOW COMPARE REGISTERS BEFORE READ DATA WITH
6464                                          ;AFTER COMMAND
6465
6466  023772  004037  031[illegible]          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6467                                                                  ;PRESENT VALUE
6468  023776  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6469  024000  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6470  024002  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6471  024004  024010                          18                      ;RETURN TO 18 ON ERROR
6472  024006  024014                          28                      ;RETURN TO 28 ON NO ERROR
6473
6474  024010  104033                  18:     ERROR   33              ;READ DATA CAUSED IMPROPER REGISTER
6475  024012  000207                          RTS     PC              ;CHANGE
6476                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
6477                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
6478                                          ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
6479                                          ;WAS GOOD
6480  024014                          28:
6481
6482  024014  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
6483  024020  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
6484  024022  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
6485  024024  000400                          256.                    ;256. WORDS TO BE COMPARED
6486  024026  024032                          38                      ;RETURN TO 38 ON ERROR
6487  024030  024036                          48                      ;RETURN TO 48 ON NO ERROR
6488
6489
6490  024032  104034                  38:     ERROR   34              ;READ DATA ERROR AFTER WRITING INTO
6491  024034  000207                          RTS     PC              ;RHAS DURING READ
6492
6493  024036                          48:
6494
6495                                          ;*************************************************************************
6496                                          ;*TEST 31       ILLEGAL FUNCTION BIT #0 IN RHER1
6497
```

```
                                        ;*      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

                                        ;*************************************************************************
024036  000004                  TST31:  SCOPE
024040  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024044  012737  000031  004470          MOV     #31,@#TSTNM     ;SAVE TEST NUMBER

024052  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER

                                        ;GENERATE ILLEGAL FUNCTION
024056  005037  001172                  CLR     @#TMP1          ;GET READY TO MAKE ILLEGAL FUNCTION
024062  012700  002310          1$:     MOV     #FUTABL,R0      ;GET POINTER TO BEGINNING OF COMMANDS
024066  012705  000021                  MOV     #17.,R5         ;COUNTER (17 GOOD FUNCTIONS)
024072  023720  001172          2$:     CMP     @#TMP1,(R0)+    ;IS THIS A LEGAL FUNCTION
024076  001004                          BNE     3$              ;BRANCH IF NOT LEGAL
024100  062737  000002  001172          ADD     #2,@#TMP1       ;MAKE ANOTHER FUNCTION
024106  000765                          BR      1$              ;GET READY TO TEST NEW FUNCTION
024110  005305                  3$:     DEC     R5              ;NOT LEGAL SO DECREMENT COUNTER
024112  001367                          BNE     2$              ;BRANCH IF 17 NOT DONE
024114  032737  000100  001172          BIT     #100,@#TMP1     ;ALL BITS UP TO BIT #5 COMPARED?
024122  001001                          BNE     20$             ;BRANCH OUT IF DONE
024124  000402                          BR      19$             ;BRANCH TO CONTINUE
024126  000137  024642          20$:    JMP     @#7$            ; DONE
024132  013737  001172  002352  19$:    MOV     @#TMP1,@#ILLEGL ;AN ILLEGAL FUNCTION IS FOUND
024140  062737  000002  001172          ADD     #2,@#TMP1       ;GET READY FOR NEW FUNCTION NEXT TIME
                                        ;ILLEGAL FUNCTION HAS BEEN FOUND
                                        ;IT IS IN 'ILLEGL'
024146  012737  024154  001110          MOV     #4$,@#LPERR     ;ERROR RETURN POINT
                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
024154                          4$:

024154  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
024160  005077  156000                  CLR     @RHWC           ;CLEAR WORD COUNT
024164  005077  155776                  CLR     @RHBA           ;CLEAR BUS ADDRESS
024170  023727  002352  000050          CMP     @#ILLEGL,#50    ;50 AND HIGHER FUNCTIONS ARE DATA
                                                                ;FUNCTIONS WHICH WILL SET MXF AND TRE
024176  103014                          BHIS    13$             ;BRANCH IF ILLEGL IS HIGHER THAN 50
024200  012737  100000  024546          MOV     #SC,@#11$+12    ;EXPECTED VALUE OF RHCS1 SHOULD HAVE
                                                                ;ONLY SC ADDED
024206  005037  024570                  CLR @#12$+12            ;EXPECTED VALUE OF RHCS2 SHOULD HAVE
                                                                ;NOTHING ADDED
024212  005037  024574                  CLR     @#12$+16        ;NO BITS TO BE CLEARED IN RHCS2
024216  005037  024604                  CLR     @#15$+6         ;RHBA SHOULD BE 0
024222  005037  024614                  CLR     @#16$+6         ;CLEAR SAVED RHWC
024226  000500                          BR      14$             ;BRANCH
024230  022737  000064  002352  13$:    CMP     #64,@#ILLEGL    ;IS FUNCTION 64
```

```
6554  024236  001020                          BNE     17$             ;BRANCH IF NOT
6555  024240  012737  140000  024546          MOV     #SC!TRE,@#11$+12;SAVED RHCS1 SHOULD HAVE SC AND TRE
6556  024246  012737  000204  024604          MOV     #204,@#15$+6    ;RHBA SHOULD HAVE 204
6557  024254  012737  000102  024614          MOV     #102,@#16$+6    ;RHWC SHOULD HAVE 102
6558  024262  012737  001200  024570          MOV     #MXF!OR,@#12$+12;RHCS2 SHOULD HAVE MXF AND OR
6559  024270  012737  000100  024574          MOV     #IR,@#12$+16    ;RHCS2 SHOULD HAVE IR CLEARED
6560  024276  000454                          BR      14$             ;BRANCH
6561  024300  022737  000066  002352  17$:     CMP     #66,@#ILLEGL    ;IS FUNCTION 66
6562  024306  001030                          BNE     18$             ;BRANCH IF NOT
6563  024310  012777  177672  155646          MOV     #-70.,@RHWC     ; MOVE 70 INTO RHWC
6564  024316  012777  002354  155642          MOV     #WRFROM,@RHBA   ;FILL RHBA WITH WRFROM
6565  024324  012737  140000  024546          MOV     #SC!TRE,@#11$+12;SAVED RHCS1
6566  024332  012767  002150  000244          MOV     #WRFROM-<66.*2>,15$+6;RHBA
6567  024340  012767  177774  000246          MOV     #-4.,16$+6      ;SAVED RHWC
6568  024346  012737  001200  024570          MOV     #MXF!OR,@#12$+12        ;SAVED RHCS2
6569  024354  005037  024574                  CLR     @#12$+16        ;RHCS2
6570  024360  012737  000100  024574          MOV     #IR,@#12$+16    ;RHCS2 SHOULD HAVE IR CLEARED
6571  024366  000420                          BR      14$                     ;BRANCH
6572  024370  005077  155570          18$:     CLR     @RHWC           ;CLEAR RHWC
6573  024374  005077  155566                  CLR     @RHBA           ;CLEAR RHBA
6574  024400  012737  140000  024546          MOV     #SC!TRE,@#11$+12;RHCS1 SHOULD HAVE SC AND TRE
6575  024406  005037  024604                  CLR     @#15$+6         ;RHBA
6576  024412  005037  024614                  CLR     @#16$+6         ;RHWC
6577  024416  012737  001000  024570          MOV     #MXF,@#12$+12   ;RHCS2
6578  024424  005037  024574                  CLR     @#12$+16        ;RHCS2
6579  024430                          14$:
6580  024430  004037  030220                  JSR     R0,@#SAVER      ;SAVE REGISTERS
6581  024434  002164                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6582  024436  004476                          SAVERE                  ;STARTING ADDRES OF WHERE
6583                                                                  ;THE REGISTERS ARE SAVED
6584  024440  000022                          10.                     ;NUMBER OF REGISTERS
6585                                                                  ;SAVED = 10.
6586
6587  024442  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6588
6589  024446  013746  002352                  MOV     @#ILLEGL,-(SP)  ;GET ILLEGAL FUNCTION
6590  024452  052716  000101                  BIS     #GO!IE,(SP)     ;INCLUDE IE AND GO
6591  024456  012611                          MOV     (SP)+,@R1       ;GO TO RHCS1 WITH ILLEGAL FUNCTION
6592
6593  024460  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
6594  024462  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6595  024464  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6596  024466  001614                          900.                    ;ALLOW 9000 MICRO SECONDS
6597  024470  001613                          907.                    ;RDY MUST SET BETWEEN
6598                                                                  ;10 AND 10150 MICRO SECONDS
6599
6600                                  ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
6601
6602  024472  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6603  024476  002174                          RHER1                   ;CHANGE RHER1 REGISTER
6604
6605  024500  000001                          1                       ;1 BIT/BITS TO BE CHANGED
6606  024502  000001                          1                       ;NEW VALUE OF ILF IS 1
6607  024504  000001                          ILF                     ;CHANGE ILF BIT
6608
6609  024506  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
```

```
6610  024512  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
6611
6612  024514  000002                          2                       ;2 BIT/BITS TO BE CHANGED
6613  024516  000001                          1                       ;NEW VALUE OF ATA IS 1
6614  024520  100000                          ATA                     ;CHANGE ATA BIT
6615  024522  000001                          1                       ;NEW VALUE OF ERR IS 1
6616  024524  040000                          ERR                     ;CHANGE ERR BIT
6617
6618  024526  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
6619                                                                  ;FOR WORKING DRIVE IN
6620                                                                  ;SAVED RHAS LOACTION
6621                                          ;RHCS1 WILL HAVE SC AND TRE ADDED IF FUNCTION IS GREATER THAN 50
6622  024534                          118:
6623
6624  024534  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6625  024540  002172                          RHCS1                   ;CHANGE RHCS1 REGISTER
6626
6627  024542  000001                          1                       ;1 BIT/BITS TO BE CHANGED
6628  024544  000001                          1                       ;NEW VALUE OF SC IS 1
6629  024546  100000                          SC                      ;CHANGE SC BIT
6630  024550  053737  002352  004504          BIS     @#ILLEGL,@#SAVERE+6;INCLUDE ILLEGAL FUNCTION
6631                                                                  ;IN RHCS1
6632                                          ;RHCS2 WILL HAVE NOTHING ADDED IF FUNCTION IS LESS THAN 50
6633  024556                          128:
6634
6635  024556  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6636  024562  002170                          RHCS2                   ;CHANGE RHCS2 REGISTER
6637
6638  024564  000002                          2                       ;2 BIT/BITS TO BE CHANGED
6639  024566  000001                          1                       ;NEW VALUE OF MXF IS 1
6640  024570  001000                          MXF                     ;CHANGE MXF BIT
6641  024572  000000                          0                       ;NEW VALUE OF IR IS 0
6642  024574  000100                          IR                      ;CHANGE IR BIT
6643  024576                          158:
6644  024576  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHBA
6645  024602  002166                          RHBA                    ;SAVED REGISTER TO CHANGE
6646  024604  000000                          0                       ;DATA
6647  024606                          168:
6648  024606  004037  027762                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6649  024612  002164                          RHWC                    ;SAVED REGISTER TO CHANGE
6650  024614  000000                          0                       ;DATA
6651
6652                                          ;NOW COMPARE REGISTERS AFTER GIVING AN ILLEGAL COMMAND
6653
6654  024616  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6655                                                                  ;PRESENT VALUE
6656  024622  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6657  024624  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6658  024626  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6659  024630  024634                          5$                      ;RETURN TO 5$ ON ERROR
6660  024632  024640                          6$                      ;RETURN TO 6$ ON NO ERROR
6661
6662  024634  104051                  5$:     ERROR   51              ;GIVING ILLEGAL FUNCTION CAUSED
6663  024636  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
6664                                                                  ;GOOD DATA GIVES WHAT
6665                                                                  ;SHOULD BE THERE
```

```
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER ILLEGAL
                                                        ;FUNCTION WA GIVEN
024640  000240                  6$:     NOP
                                                ;JMP    @#1$;BRANCH FOR NEXT FUNCTION
024642                            7$:
024642                            10$:
                                ;*****************************************************************
                                ;*TEST 32       ERROR REGISTER 1 - BIT #13 OPI
                                ;*      A WRITE HEADER AND DATA COMMAND IS GIVEN
                                ;*      CYLINDER 0 SECTOR 1 TRACK 0 KEYS 0 DATA 177777
                                ;*      WORDCOUNT 260
                                ;*
                                ;*      AFTER GO IS GIVEN THEN THREE INDEX PULSES ARE
                                ;*      GIVEN. THIS SHOULD BRING OPI HIGH
                                ;*****************************************************************
024642  000004                  TST32:  SCOPE
024644  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024650  012737  000032  004470          MOV     #32,@#TSTNM     ;SAVE TEST NUMBER
024656  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                ;THESE ARE REGULAR SETUPS
024662  012777  177374  155274          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
024670  012700  002354                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
024674  010077  155266                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
024700  012710  010000                  MOV     #FMT22,(R0);    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
024704  012720  000001                  MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
024710  005020                          CLR     (R0)+           ;KEY1=0
024712  005020                          CLR     (R0)+           ;KEY2=0
024714  012705  000400                  MOV     #256.,R5        ;COUNTER
024720  012720  177777          18:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
024724  005305                          DEC     R5
024726  001374                          BNE     18              ;BRANCH IF DATA NOT COMPLETE
024730  012777  000001  155240          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
024736  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
024742  013711  002332                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
                                                                ;DATA WITH 62 IN RHCS1
024746  005037  004616                  CLR     @#ERFLG#        ;CLEAR ERROR FLAG
024752  012777  010000  155222          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
024760  005077  155220                  CLR     @RHCA           ;CYLINDER =0
```

```
6722
6723                                    ;SAVE REGISTERS FOR COMPARAISON AFTER READ
6724  024764  004037  030220            JSR     R0,@#SAVER      ;SAVE REGISTERS
6725  024770  002164                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6726  024772  004476                    SAVERF                  ;STARTING ADDRES OF WHERE
6727                                                            ;THE REGISTERS ARE SAVED
6728  024774  000023                    19.                     ;NUMBER OF REGISTERS
6729                                                            ;SAVED = 19.
6730
6731                                    ;GO TO WRITE HEADER AND DATA
6732                                    ;BUT BEFORE GO ONE INDEX PULSE IS GIVEN
6733                                    ;TO CLEAR OUT THE SECTOR CLOCK COUNTER
6734                                    ;SO THAT NO SECTOR PULES COMES IN THIS TEST
6735
6736  024776  013700  002212            MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
6737  025002  012710  000001            MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
6738  025006  052710  000004            BIS     #MINX,@R0       ;SET INDEX
6739  025012  042710  000004            BIC     #MINX,@R0       ;CLEAR INDEX THIS GIVES
6740                                                            ;ONE INDEX PULSE
6741
6742  025016  052777  000001  155146    BIS     #GO,@RHCS1      ;GO
6743
6744
6745
6746                            ;WAIT FOR OR BIT IN RHCS2  TIME IS NOT IMPORTANT
6747
6748  025024  104420                    WAT                     ;WAIT FOR OR BIT TO SET
6749  025026  002170                    RHCS2                   ;WAIT FOR RHCS2 REGISTER
6750  025030  000200                    OR                      ;WAIT FOR OR BIT IN RHCS2 REGISTER
6751  025032  001604                    900.                    ;ALLOW 9000 MICRO SECONDS
6752  025034  001604                    900.                    ;OR MUST SET BETWEEN
6753                                                            ;00 AND 18000 MICRO SECONDS
6754
6755
6756  025036  052710  000004            BIS     #MINX,@R0       ;SET INDEX  CLOCK
6757  025042  042710  000004            BIC     #MINX,@R0       ;RESET INDEX CLOCK
6758
6759                                    ;SECOND INDEX PULSE
6760  025046  052710  000004            BIS     #MINX,@R0       ;SET INDEX
6761  025052  042710  000004            BIC     #MINX,@R0       ;CLEAR INDEX
6762
6763                                    ;THIRD INDEX PULSE
6764  025056  052710  000004            BIS     #MINX,@R0       ;SET INDEX
6765  025062  042710  000004            BIC     #MINX,@R0       ;CLEAR INDEX
6766
6767
6768                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
6769
6770  025066  004037  030732            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6771  025072  002172                    RHCS1                   ;CHANGE RHCS1 REGISTER
6772
6773  025074  000002                    2                       ;2 BIT/BITS TO BE CHANGED
6774  025076  000001                    1                       ;NEW VALUE OF SC IS 1
6775  025100  100000                    SC                      ;CHANGE SC BIT
6776  025102  000001                    1                       ;NEW VALUE OF TRE IS 1
6777  025104  040000                    TRE                     ;CHANGE TRE BIT
```

```
6778
6779  025106  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6780  025112  002214                          RHDS1                   ;CHANGE RHDS1 REGISTER
6781
6782  025114  000002                          2                       ;2 BIT/BITS TO BE CHANGED
6783  025116  000001                          1                       ;NEW VALUE OF ATA IS 1
6784  025120  100000                          ATA                     ;CHANGE ATA BIT
6785  025122  000001                          1                       ;NEW VALUE OF ERR IS 1
6786  025124  040000                          ERR                     ;CHANGE ERR BIT
6787  025126  004037  027762                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
6788  025132  002176                          RHDST                   ;SAVED REGISTER TO CHANGE
6789  025134  000002                          2                       ;DATA
6790
6791  025136  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6792  025142  002174                          RHER1                   ;CHANGE RHER1 REGISTER
6793
6794  025144  000001                          1                       ;1 BIT/BITS TO BE CHANGED
6795  025146  000001                          1                       ;NEW VALUE OF OPI IS 1
6796  025150  020000                          OPI                     ;CHANGE OPI BIT
6797
6798  025152  053737  004622  004522          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPIATE ATA BITS
6799                                                                  ;FOR WORKING DRIVE IN
6800                                                                  ;SAVED RHAS LOACTION
6801
6802  025160  004037  030732                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
6803  025164  002212                          RHMR                    ;CHANGE RHMR REGISTER
6804
6805  025166  000001                          1                       ;1 BIT/BITS TO BE CHANGED
6806  025170  000001                          1                       ;NEW VALUE OF DMD IS 1
6807  025172  000001                          DMD                     ;CHANGE DMD BIT
6808
6809
6810                                          ;RHWC,RHBA AND OR AND IR BITS OF RHCS2 WILL NOT BE CHECKED
6811  025174  017737  154764  004476          MOV     @RHWC,@#SAVERE  ;SAVED RHWC
6812  025202  017737  154760  004500          MOV     @RHBA,@#SAVERE+2        ;SAVED RHBA
6813  025210  017746  154754                  MOV     @RHCS2,-(SP)    ;GET RHCS2
6814  025214  042716  177477                  BIC     #^C<IR!OR>,(SP) ;GET IR OR STATES
6815  025220  042737  000300  004502          BIC     #IR!OR,@#SAVERE+4       ;CLEAR IR OR
6816  025226  052637  004502                  BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
6817
6818                                          ;COMPARE REGISTERS BEFORE WRITE WITH AFTER ATTEMPT
6819
6820  025232  004037  031040                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6821                                                                  ;PRESENT VALUE
6822  025236  004476                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6823  025240  002242                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6824  025242  000021                          17.                     ;17. REGISTERS TO BE COMPARED
6825  025244  025250                          2$                      ;RETURN TO 2$ ON ERROR
6826  025246  025254                          3$                      ;RETURN TO 3$ ON NO ERROR
6827
6828  025250  104071                  2$:     ERROR   71              ;FORCING OPI
6829  025252  000207                          RTS     PC              ;CAUSED
6830                                                                  ;IMPROPER REGISTER CHANGE
6831                                                                  ;GOOD DATA GIVES WHAT SHOULD BE
6832                                                                  ;THERE
6833                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
```

```
                                                        ;AFTER 3 INDEX PULSES

025254  004737  030060          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT


                        ;*********************************************************************
                        ;*TEST 33       ERROR REGISTER #1, 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.

                        ;*********************************************************************
025260  000004          TST33:  SCOPE
025262  012706  001000          MOV     #STACK,SP       ;RESET STACK
025266  012737  000033  004470  MOV     #33,@#TSTNM     ;SAVE TEST NUMBER

025274  004737  030060          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUBER
025300  012737  000025  025334  MOV     #21.,@#18+12            ;SET UP TO START FROM
025306  012737  000025  025350  MOV     #21.,@#28+6             ;SECTOR 21.
025314  012737  000056  004630  MOV     #46.,@#TMP1             ;46 SECTORS TO COVER 3 TRACKS
                        ;FILL WRITE FROM BUFFER WITH HEADER
025322                  1$:

025322  004037  027704          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
025326  002354          WRFROM                          ;LOCATION WHERE SAVED
025330  000004          4                               ;NUMBER OF WORDS SAVED
025332  010000          10000                           ;FIRST DATA WORD
025334  000025          21.                             ;SECOND DATA WORD
025336  000000          0                               ;THIRD DATA WORD
025340  000000          0                               ;FOURTH DATA WORD

                        ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
025342                  2$:
```

```
6890  025342  004037  032024              JSR     R0,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
6891  025346  000000                      0                         ;CYLINDER 0
6892  025350     025                .BYTE  21.                       ;SECTOR 21.
6893  025351     000                .BYTE  0                         ;TRACK 0
6894  025352  177774                      -0-4              ;WORD COUNT (DATA)=0+
6895                                                        ;4 HEADER WORDS
6896  025354  002354                      WRFROM                    ;BUS ADDRESS
6897                                                        ;STARTING ADDRESS OF DATA
6898                                                        ;BUFFER = WRFROM
6899  025356  000000                      0                         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6900  025360  010000                      FMT22                     ;16 BITS PER WORD FORMAT
6901                                                        ;DO NOT INHIBIT ECC CORRECTION
6902                                                        ;DO NOT INHIBIT HEADER COMPARE
6903  025362  002332                      WRIFOR                    ;GET READY TO DO A WRIFOR
6904                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1
6905
6906
6907  025364  004737  030140              JSR     PC,@#CHECKT       ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6908
6909
6910  025370  013777  004472  154562      MOV     @#RP4VEC,@RPVEC   ;SET RP04 VECTOR ADDRESS
6911                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
6912                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6913                                                        ;'TIME' WILL ONLY SAVE
6914                                                        ;CURRENT CYLINDER ADDRESS
6915                                                        ;AND LOOK AHEAD REGISTERS
6916
6917
6918  025376  013746  002332              MOV     @#WRIFOR,-(SP)    ;GET READY TO MOVE COMMAND
6919  025402  052716  000101              BIS     #GO!IE,(SP)       ;GET READY TO SET GO AND
6920                                                        ;ENABLE INTERRUPT
6921  025406  012677  154560              MOV     (SP)+,@RHCS1      ;GO WITH
6922                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
6923                                                        ;WITH INTERRUPT ENABLED
6924  025412  011100                      MOV     @R1,R0            ;SAVE RHCS1 DURING ABOVE OPERATION
6925  025414  011305                      MOV     @R3,R5            ;SAVE RHDS1 DURING ABOVE OPERATION
6926
6927                                      ;ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6928
6929  025416  104420                      WAT                       ;WAIT FOR RDY BIT TO SET
6930  025420  002172                      RHCS1                     ;WAIT FOR RHCS1 REGISTER
6931  025422  000200                      RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6932  025424  003237                      1695.                     ;ALLOW 16950 MICRO SECONDS
6933  025426  001515                      845.                      ;RDY MUST SET BETWEEN
6934                                                        ;8500 AND 25400 MICRO SECONDS
6935
6936                                      ;NOW ONE MORE SECTOR HAS BEEN WRITTEN
6937                                      ;SC WILL BE CHECKED TO MAKE SURE
6938                                      ;NO ERRORS OCCURED
6939  025430  017737  154536  002250      MOV     @RHCS1,@#CS1              ;GET RHCS1
6940  025436  032737  100000  002250      BIT     #SC,@#CS1                 ;IS SC SET
6941  025444  001403                      BEQ     3$                        ;BRANCH IF NO
6942  025446  004737  031764              JSR     PC,@#PUTREG               ;SAVE REGISTERS
6943  025452  104072                      ERROR   72                        ;THERE WAS AN ERROR AFTER
6944                                                                        ;A WRITE HEADER AND DATA.
6945                                      ;ONE SECTOR HAS BEEN FORMATTED NOW
```

```
                                        ;CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
                                        ;
                                        ;
025454  013705  025334          3$:     MOV     @#1$+12,R5          ;GET SECTOR TRACK WORD
025460  005205                          INC     R5
025462  122705  000026                  CMPB    #22.,R5             ;IS IT 22
025466  001405                          BEQ     4$                  ;BRANCH IF 21 IS DONE
025470  010537  025334                  MOV     R5,@#1$+12          ;RESTORE SECTOR TRACK
025474  010537  025350                  MOV     R5,@#2$+6           ;RESTORE SECTOR TRACK
025500  000410                          BR      5$
025502  105037  025334          4$:     CLRB    @#1$+12             ;RESTORE SECTOR
025506  105237  025335                  INCB    @#1$+13             ;RESTORE TRACK
025512  105037  025350                  CLRB    @#2$+6              ;RESTORE SECTOR
025516  105237  025351                  INCB    @#2$+7              ;RESTORE TRACK
025522  005337  004630          5$:     DEC     @#TMP1              ;IS 46 SECTORS DONE
025526  001275                          BNE     1$                  ;BRANCH IF NOT
                                        ;
                                        ;
                                        ;NOW 46 SECTORS HAVE BEEN FORMATTED
                                        ;READ HEADER AND DATA FOR 46 SECTORS=11960 WORDS
                                        ;WITH BUS ADDRESS INHIBITED
                                        ;
                                        ;
025530  004737  030060                  JSR     PC,@#CLDISK         ;SET R1-RHCS1, R2-RHCS2
                                                                    ;R3-RHDS1, R4-RHER1
                                                                    ;GIVE RH-11 INITIALIZE
                                                                    ;SETUP UNIT NUBER
                                        ;FILL READ HEADER AND DATA COMMAND
                                        ;
025534  004037  032024                  JSR     R0,@#RUN            ;SETUP TO RUN FOR DATA COMMAND
025540  000000                          0                           ;CYLINDER 0
025542     025                  .BYTE   21.                         ;SECTOR 21.
025543     000                  .BYTE   0                           ;TRACK 0
025544  150510                          -11956.-4                   ;WORD COUNT (DATA)=11956.+
                                                                    ;4 HEADER WORDS
025546  003420                          REINTO                      ;BUS ADDRESS
                                                                    ;STARTING ADDRESS OF DATA
                                                                    ;BUFFER = REINTO
025550  000010                          BAI                         ;INHIBIT BUS ADDRESS INCREMENT
025552  014000                          FMT22!ECI                           ;16 BITS PER WORD FORMAT
                                                                    ;INHIBIT ECC CORRECTION
                                                                    ;DO NOT INHIBIT HEADER COMPARE
025554  002336                          REFOR                       ;GET READY TO DO A REFOR
                                                                    ;READ HEADER AND DATA WITH 72 IN RHCS1
                                        ;
                                        ;
025556  004737  030140                  JSR     PC,@#CHECKT         ;CHECK DVA,RDY,MOL,DPR,DRY,VV
                                        ;
                                        ;
025562  013777  004472  154370          MOV     @#PP4VEC,@RPVEC     ;SET RP04 VECTOR ADDRESS
                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                    ;'TIME' WILL ONLY SAVE
                                                                    ;CURRENT CYLINDER ADDRESS
                                                                    ;AND LOOK AHEAD REGISTERS
                                        ;
```

```
7002
7003  025570  013746  002336                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
7004  025574  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7005                                                                    ;ENABLE INTERRUPT
7006  025600  012677  154366                  MOV     (SP)+,@RHCS1    ;GO WITH
7007                                                                      ;72 IN RHCS1 FOR READ DATA
7008                                                                    ;WITH INTERRUPT ENABLED
7009  025604  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7010  025606  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7011                                          ;TIME IS NOT IMPORTANT
7012
7013  025610  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
7014  025612  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
7015  025614  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7016  025616  121320                          41680.                  ;ALLOW 416800 MICRO SECONDS
7017  025620  121320                          41680.                  ;RDY MUST SET BETWEEN
7018                                                                    ;00 AND 178240 MICRO SECONDS
7019
7020                                          ;NOW THAT ALL 11960 WORDS HAVE BEEN READ
7021                                          ;SC WILL BE CHECKED FOR NO ERRORS
7022  025622  017737  154344  002250          MOV     @RHCS1,@#CS1            ;GET RHCS1
7023  025630  032737  100000  002250          BIT     #SC,@#CS1               ;IS SC SET
7024  025636  001403                          BEQ     6$                      ;BRANCH IF NO
7025  025640  004737  031764                  JSR     PC,@#PUTREG             ;SAVE REGISTERS
7026  025644  104073                          ERROR   73                      ;READ HEADER AND DATA
7027                                                                          ;FOR 11960 WORDS
7028                                                                          ;THAT IS OVER THREE
7029                                                                          ;INDEX PULSES CAUSED
7030                                                                          ;AN ERROR
7031
7032  025646  017737  154322  002252  6$:     MOV     @RHER1,@#ER1            ;GET RHER1
7033  025654  032737  020000  002252          BIT     #OPI,@#ER1              ;IS OPI SET
7034  025662  001403                          BEQ     7$                      ;BRANCH IF NO
7035  025664  004737  031764                  JSR     PC,@#PUTREG             ;SAVE REGISTERS
7036  025670  104074                          ERROR   74                      ;READ HEADER AND DATA
7037                                                                          ;OVER 3 INDEX PULSES
7038                                                                          ;CAUSED OPI TO SET
7039  025672                          7$:
7040
7041
7042
7043
7044
7045
7046                                  ;*************************************************************************
7047                                  ;*TEST 34        HEAD SELECTION TEST
7048                                  ;*      THIS TESTS HEAD SELECTION LOGIC ONLY. A WRITE HEADER AND
7049                                  ;*      DATA COMMAND IS GIVEN TO EACH TRACK FROM 0 TO 10 ON
7050                                  ;*      CYLINDER 0, SECTOR 0.
7051                                  ;*      THE DATA ON EACH SECTOR IS UNIQUE. THE LEAST SIGNIFICANT
7052                                  ;*      5 BITS GIVE SECTOR THE NEXT LEAST SIGNIFICANT 5 BITS
7053                                  ;*      GIVE TRACK THE NEXT 6 BITS GIVE CYLINDER
7054                                  ;*
7055                                  ;*      THEN READ HEADER AND DATA IS DONE FOR THE ABOVE AND DATA
7056                                  ;*      CHECKED
7057                                  ;*
```

```
                                        ;*      BETWEEN THE WRITE AND READ ONLY ERR AND TRE ARE CHECKED
                                        ;*
                                        ;*
                                        ;*      ON AN ERROR IN THE READ HEADER AND DATA LOOPING WILL BE
                                        ;*      ONLY ON THE ERROR SECTOR READ
                                        ;
                                        ;*************************************************************
025672  000004                  TST34:  SCOPE
025674  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025700  012737  000034  004470          MOV     #34,@#TSTNM     ;SAVE TEST NUMBER
                                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                        ;THE FOLLOWING CLEARS ARE TO INITIALIZE TEST FROM CYLINDER 0
025712  005037  026012              CLR     @#1$+12         ;START WITH SECTOR/TRACK = 0
025716  005037  026030              CLR     @#2$+10         ;START WITH DATA = 0
025722  005037  026040              CLR     @#3$+6          ;START WITH 0 FOR COMMAND
025726  012737  000023  001172      MOV     #19.,@#STMP1    ;19 TRACKS TO BE WRITTEN
                                        ;THIS GETS THE HEADS TO CYLINDER 0
025734  004737  030140              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
025740  013777  004472  154212      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CUPRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS
025746  013746  002314              MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
025752  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
025756  012677  154210              MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;6 IN RHCS1 FOR RECALIBRATE
                                                                ;WITH INTERRUPT ENABLED
025762  104420                      WAT                     ;WAIT FOR DRY BIT TO SET
025764  002214                      RHDS1                   ;WAIT FOR RHDS1 REGISTER
025766  000200                      DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
025770  060650                      25000.                  ;ALLOW 250000 MICRO SECONDS
025772  060650                      25000.                  ;DRY MUST SET BETWEEN
                                                                ;00 AND 500000 MICRO SECONDS
025774  004737  030060              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                        ;FILL WRITE FROM BUFFER WITH HEADER
026000                          1$:
```

```
7114  026000  004037  027704                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
7115  026004  002354                          WRFROM                  ;LOCATION WHERE SAVED
7116  026006  000004                          4                       ;NUMBER OF WORDS SAVED
7117  026010  010000                          10000                   ;FIRST DATA WORD
7118  026012  000000                          <0*400>!0                       ;SECOND DATA WORD
7119  026014  000000                          0                       ;THIRD DATA WORD
7120  026016  000000                          0                       ;FOURTH DATA WORD
7121
7122                                          ;FILL WRITE FROM BUFFER WITH DATA
7123  026020                          2$:
7124  026020  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
7125  026024  002364                          WRFROM+10                       ;STARTING FROM WRFROM+10
7126  026026  000400                          256.                    ;256. WORDS
7127  026030  000000                          <0.*2000>!<0.*40>!0                     ;FILL WITH <0.*2000>!<0.*40>!0
7128
7129
7130                                          ;THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7131  026032                          3$:
7132
7133  026032  004037  032024                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7134  026036  000000                          0                       ;CYLINDER 0
7135  026040     000          .BYTE   0                       ;SECTOR 0
7136  026041     000          .BYTE   0                       ;TRACK 0
7137  026042  177374                          -256.-4         ;WORD COUNT (DATA)=256.+
7138                                                          ;4 HEADER WORDS
7139  026044  002354                          WRFROM                  ;BUS ADDRESS
7140                                                          ;STARTING ADDRESS OF DATA
7141                                                          ;BUFFER = WRFROM
7142  026046  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7143  026050  010000                          FMT22                   ;16 BITS PER WORD FORMAT
7144                                                          ;DO NOT INHIBIT ECC CORRECTION
7145                                                          ;DO NOT INHIBIT HEADER COMPARE
7146  026052  002332                          WRIFOR                  ;GET READY TO DO A WRIFOR
7147                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7148
7149
7150  026054  004737  030140                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV
7151
7152
7153  026060  013777  004472  154072          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7154                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
7155                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7156                                                          ;'TIME' WILL ONLY SAVE
7157                                                          ;CURRENT CYLINDER ADDRESS
7158                                                          ;AND LOOK AHEAD REGISTERS
7159
7160
7161  026066  013746  002332                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
7162  026072  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7163                                                          ;ENABLE INTERRUPT
7164  026076  012677  154070                  MOV     (SP)+,@RHCS1    ;GO WITH
7165                                                           ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7166                                                          ;WITH INTERRUPT ENABLED
7167
7168                                          ;ONE REVOLUTION = 16670 MICRO SEC., ONE SECTOR = 760
7169                                          ;MICRO SEC. MAX TIME ALLOWED = ONE REVOLUTION + HEAD
```

```
                                        ;SWITCH + 2 SECTORS, MIN TIME ALLOWED = SECTOR (FIRST CASE)
                                        ;IF THERE IS A FAILURE HERE HALT PROGRAM AFTER ERROR WITH
                                        ;SWITCH 15 AND SEE CURRENT CYLINDER REGISTER TO DETERMINE
                                        ;WHAT CYLINDER IS FAILING


026102  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
026104  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
026106  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
026110  003162                          1650.                   ;ALLOW 16500 MICRO SECONDS
026112  001572                          890.                    ;RDY MUST SET BETWEEN
                                                                ;7600 AND 25400 MICRO SECONDS


                                        ;NOW SECTOR 0 OF ONE TRACK HAS BEEN WRITTEN CHECK COMPOSIT
                                        ;ERROR BIT TO BE SURE NO ERRORS HAPPENED

                                        ;SAVE REGISTERS IN SAVE TABLE
026114  004737  031764                  JSR     PC,@#PUTREG

026120  032737  040000  002272          BIT     #ERR,@#DS1      ;ANY DISK ERRORS
026126  001004                          BNE     9$              ;BRANCH IF YES
026130  032737  040000  002250          BIT     #TRE,@#CS1      ;ANY RH ERRORS
026136  001401                          BEQ     4$              ;BRANCH IF NO

026140  104066                  9$:     ERROR   66              ;SOME ERRORS OCCURRED
                                                                ;WHILE DOING WRITE HEADER
                                                                ;AND DATA


                                        ;THE FOLLOWING 3 ADDS SETS UP FOR NEXT TRACK WRITING

026142  062737  000400  026012  4$:     ADD     #400,@#1$+12    ;NEXT TRACK FOR HEADER
026150  062737  000040  026030          ADD     #40,@#2$+10     ;NEXT TRACK FOR DATA
026156  062737  000400  026040          ADD     #400,@#3$+6     ;NEXT TRACK FOR COMMAND

026164  005337  001172                  DEC     @#TMP1          ;COUNT 19 TRACKS
026170  001303                          BNE     1$

                                        ;THE FOLLOWING CLEARS SETS UP FOR READ HEADER AND DATA
026172  005037  026252                  CLR     @#SST3+12               ;START WITH SECTOR/TRACK = 0
026176  005037  026270                  CLR     @#SST4+10               ;START WITH DATA = 0
026202  005037  026300                  CLR     @#SST5+6                ;START WITH 0 FOR COMMAND


026206  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUBER
                                        ;SET UP FOR READ HEADER AND DATA
026212  012737  000023  001172  SST1:   MOV     #19.,@#TMP1     ;19 TRACKS TO BE READ

                                        ;FILL READ INTO BUFFER WITH ALL ONES
026220                          SST2:
026220  004037  027730                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
026224  003420                          REINTO                  ;STARTING FROM REINTO
026226  000404                          260.                    ;260. WORDS
026230  177777                          -1                      ;FILL WITH -1
```

```
                                ;SET LOOP POINT
                                ;FILL WRITE FROM BUFFER WITH EXPECTED HEADER
SST3:
                                ;SAVE HEADER DATA IN WRFROM
                                ;LOCATION WHERE SAVED
                                ;NUMBER OF WORDS SAVED
                                ;FIRST DATA WORD
                                ;SECOND DATA WORD
                                ;THIRD DATA WORD
                                ;FOURTH DATA WORD
                                ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
SST4:
                                ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
                                        ;STARTING FROM WRFROM+<4*2>
                                ;256. WORDS
                                                        ;FILL WITH <0,*2000>!<0*40>!0
                                ;FILL COMMAND FOR READ HEADER AND DATA
SST5:
                                ;SETUP TO RUN FOR DATA COMMAND
                                ;CYLINDER 0
                                ;SECTOR 0
                                ;TRACK 0
                                ;WORD COUNT (DATA)=256.+
                                ;4 HEADER WORDS
                                ;BUS ADDRESS
                                ;STARTING ADDRESS OF DATA
                                ;BUFFER = REINTO
                                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
                                        ;16 BITS PER WORD FORMAT
                                ;INHIBIT ECC CORRECTION
                                ;DO NOT INHIBIT HEADER COMPARE
                                ;GET READY TO DO A REFOR
                                ;READ HEADER AND DATA WITH 72 IN RHCS1


                                ;CHECK DVA,RDY,MOL,DPR,DRY,VV


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


                                ;GET READY TO MOVE COMMAND
                                ;GET READY TO SET GO AND
                                ;ENABLE INTERRUPT
```

```
7282  026336  012677  153630                  MOV     (SP)+,@#RHCS1   ;GO WITH
7283                                                                    ;72 IN RHCS1 FOR READ DATA
7284                                                                    ;WITH INTERRUPT ENABLED
7285
7286
7287  026342  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
7288  026344  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
7289  026346  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7290  026350  003162                          1650.                   ;ALLOW 16500 MICRO SECONDS
7291  026352  001572                          890.                    ;RDY MUST SET BETWEEN
7292                                                                    ;7600 AND 25400 MICRO SECONDS
7293
7294                                          ;NOW SECTOR 0 OF ONE TRACK HAS BEEN READ CHECK COMPOSIT
7295                                          ;ERROR BIT TO BE SURE NO ERROR HAPPENED
7296
7297                                          ;SAVE REGISTERS IN SAVE TABLE
7298  026354  004737  031764                  JSR     PC,@#PUTREG
7299
7300  026360  032737  040000  002272          BIT     #ERR,@#DS1      ;ANY DISK ERRORS
7301  026366  001004                          BNE     10$             ;BRANCH IF YES
7302  026370  032737  040000  002250          BIT     #TRE,@#CS1      ;ANY RH ERRORS
7303  026376  001401                          BEQ     11$             ;BRANCH IF NO
7304
7305  026400  104066                  10$:    ERROR   66              ;SOME ERRORS OCCURRED
7306                                                                    ;WHILE DOING READ
7307                                                                    ;HEADER AND DATA
7308
7309                                          ;NOW THE READ DATA WILL BE COMPARED DATA IN EACH SECTOR
7310                                          ;IS UNIQUE IF PROGRAM IS HALTED ON ERROR THEN LOOK AT
7311                                          ;RHDST TO GET WHAT TRACK IS IN ERROR. LOOKING AT THE DATA
7312                                          ;BITS NO 4,5,6,7,8 IN GOOD DATA ALSO GIVES TRACK NUMBER
7313                                          ;IN GOOD DATA ALSO GIVES TRACK NUMBER
7314  026402                          11$:
7315
7316  026402  004037  032070                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7317  026406  002354                          WRFROM                  ;GOOD DATA      STARTS FROM WRFROM
7318  026410  003420                          REINTO                  ;TEST DATA STARTS FROM REINTO
7319  026412  000404                          260.                    ;260. WORDS TO BE COMPARED
7320  026414  026420                          12$                     ;RETURN TO 12$ ON ERROR
7321  026416  026424                          13$                     ;RETURN TO 13$ ON NO ERROR
7322
7323
7324                                                                    ;BITS 4,5,6,7,8
7325  026420  104067                  12$:    ERROP   67              ;READ HEADER AND DATA
7326  026422  000207                          RTS     PC              ;ERROR
7327                                                                    ;HEAD SELECTION ERROR
7328                                                                    ;DATA READ GIVES NATURE
7329                                                                    ;OF ERROR
7330                                                                    ;EXCEPT FOR THE
7331                                                                    ;FOUR  HEADER WORDS
7332                                                                    ;THE BITS 4,5,6,7,8
7333                                                                    ;GIVE THE TRACK NUMBER
7334
7335                                          ;NOW INCREMENT TO READ NEXT TRACK
7336
7337  026424  062737  000400  026252  13$:    ADD     #400,@#SST3+12  ;NEXT TRACK FOR HEADER
```

```
 7338  026432  062737  000040  026270          ADD     #40,@#SST4+10   ;NEXT TRACK FOR DATA
 7339  026440  062737  000400  026300          ADD     #400,@#SST5+6   ;NEXT TRACK FOR COMMAND
 7340
 7341  026446  005337  001172                  DEC     @#STMP1         ;COUNT 19 TRACKS
 7342  026452  001001                          BNE     5$
 7343
 7344  026454  000402                          BR      TST35   ;TO NEXT TEST
 7345
 7346  026456  000137  026220          5$:     JMP     @#SST2          ;JUMP BACK
 7347
 7348                                  ;*****************************************************************
 7349                                  ;*TEST 35       DIFFERENCE LINES
 7350
 7351                                  ;*      A WRITE HEADER AND DATA WILL BE DONE ON ALL CYLINDERS
 7352                                  ;*      UP TO 256 ON SECTOR ZERO, TRACK ZERO. THE DATA WILL BE THE
 7353                                  ;*      CYLINDER NUMBER
 7354                                  ;*
 7355                                  ;*      THEN A RECALIBRATE AND READ HEADER AND DATA WILL BE DONE
 7356                                  ;*      ON CYLINDERS 0,1,2,4,8,16,32,64,128,256.
 7357                                  ;*
 7358                                  ;*      DATA WILL BE CHECKED
 7359                                  ;*      ON AN ERROR LOOPING WILL BE DONE ON
 7360                                  ;*      READ ONLY
 7361                                  ;*
 7362                                  ;*****************************************************************
 7363  026462  000004                  TST35:  SCOPE
 7364  026464  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 7365  026470  012737  000035  004470          MOV     #35,@#TSTNM     ;SAVE TEST NUMBER
 7366
 7367  026476  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 7368                                                                  ;R3-RHDS1, R4-RHER1
 7369                                                                  ;GIVE RH-11 INITIALIZE
 7370                                                                  ;SETUP UNIT NUBER
 7371
 7372                                  ;THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM CYLINDER
 7373                                  ;ZERO
 7374
 7375                                  ;SET UP TO INITIALIZE TEST FROM CYLINDER 0, TRACK 0,
 7376                                  ;SECTOR 0
 7377  026502  012737  010000  026576          MOV     #10000,@#1$+10  ;CYLINDER HEADER DATA
 7378  026510  005037  026616                  CLR     @#2$+10         ;DATA
 7379  026514  005037  026624                  CLR     @#3$+4          ;CYLINDER COMMAND RHCA
 7380
 7381                                  ;THIS IS TO GET THE HEADS TO CYLINDER ZERO
 7382  026520  013777  004472  153432          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 7383                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 7384                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 7385                                                                  ;'TIME' WILL ONLY SAVE
 7386                                                                  ;CURRENT CYLINDER ADDRESS
 7387                                                                  ;AND LOOK AHEAD REGISTERS
 7388
 7389  026526  013746  002314                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
 7390  026532  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 7391                                                                  ;ENABLE INTERRUPT
 7392  026536  012677  153430                  MOV     (SP)+,@RHCS1    ;GO WITH
 7393                                                                   ;6 IN RHCS1 FOR RECALIBRATE
```

```
                                                        ;WITH INTERRUPT ENABLED
                                                        ;WAIT FOR DRY BIT TO SET
026542  104420                  WAT                   ;WAIT FOR RHDS1 REGISTER
026544  002214                  RHDS1                 ;WAIT FOR DRY BIT IN RHDS1 REGISTER
026546  000200                  DRY                   ;ALLOW 250000 MICRO SECONDS
026550  060650                  25000.                ;DRY MUST SET BETWEEN
026552  060650                  25000.                ;00 AND 500000 MICRO SECONDS
                                                      ;257 CYLINDERS
026554  012737  000401  001172  MOV     #257.,@#TMP1  ;SET R1=RHCS1, R2=RHCS2
                                                      ;R3=RHDS1, R4=RHER1
026562  004737  030060          JSR     PC,@#CLDISK   ;GIVE RH-11 INITIALIZE
                                                      ;SETUP UNIT NUBER

                                                      ;FILL WRITE FROM BUFFER WITH HEADER
026566                      1$:
                                                      ;SAVE HEADER DATA IN WRFROM
026566  004037  027704          JSR     R0,@#FLHEAD   ;LOCATION WHERE SAVED
026572  002354                  WRFROM                ;NUMBER OF WORDS SAVED
026574  000004                  4                     ;FIRST DATA WORD
026576  010000                  10000                 ;SECOND DATA WORD
026600  000000                  0                     ;THIRD DATA WORD
026602  000000                  0                     ;FOURTH DATA WORD
026604  000000                  0

                                                      ;FILL WRITE FROM BUFFER WITH DATA
026606                      2$:
                                                      ;CLEAR 256. WORDS, FROM WRFROM+10
026606  004037  027730          JSR     R0,@#CLAREA   ;STARTING FROM WRFROM+10
026612  002364                  WRFROM+10             ;256. WORDS
026614  000400                  256.                  ;FILL WITH 0
026616  000000                  0

                                                      ;THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
026620                      3$:

                                                      ;SETUP TO RUN FOR DATA COMMAND
026620  004037  032024          JSR     R0,@#RUN      ;CYLINDER 0
026624  000000                  0                     ;SECTOR 0
026626     000          .BYTE   0                     ;TRACK 0
026627     000          .BYTE   0
026630  177374                  -256.-4       ;WORD COUNT (DATA)=256.+
                                                      ;4 HEADER WORDS
026632  002354                  WRFROM                ;BUS ADDRESS
                                                      ;STARTING ADDRESS OF DATA
                                                      ;BUFFER = WRFROM
026634  000000                  0                     ;DO NOT INHIBIT BUS ADDRESS INCREMENT
026636  010000                  FMT22                 ;16 BITS PER WORD FORMAT
                                                      ;DO NOT INHIBIT ECC CORRECTION
                                                      ;DO NOT INHIBIT HEADER COMPARE
026640  002332                  WRIFOR                ;GET READY TO DO A WRIFOR
                                                ;WRITE HEADER AND DATA WITH 67 IN RHCS1

026642  004737  030140          JSR     PC,@#CHECKT   ;CHECK DVA,RDY,MOL,DPR,DRY,VV
```

```
7450
7451
7452  026646  013777  004472  153304          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7453                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
7454                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7455                                                                  ;'TIME' WILL ONLY SAVE
7456                                                                  ;CURRENT CYLINDER ADDRESS
7457                                                                  ;AND LOOK AHEAD REGISTERS
7458
7459
7460  026654  013746  002332                  MOV     @#WRIFOP,-(SP)  ;GET READY TO MOVE COMMAND
7461  026660  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7462                                                                  ;ENABLE INTERRUPT
7463  026664  012677  153302                  MOV     (SP)+,@RHCS1    ;GO WITH
7464                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7465                                                                  ;WITH INTERRUPT ENABLED
7466
7467                                          ;ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760
7468                                          ;MICRO SECONDS, ONE SEEK = 7000 MICRO SECONDS, MAX
7469                                          ;TIME = ONE REVOLUTION + SEEK + 2 SECTORS, MIN TIME =
7470                                          ;ONE SECTOR
7471
7472
7473  026670  104420                          WAT                     ;WAIT FOR RDY BIT TO SET
7474  026672  002172                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
7475  026674  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7476  026676  002354                          1260.                   ;ALLOW 12600 MICRO SECONDS
7477  026700  002354                          1260.                   ;RDY MUST SET BETWEEN
7478                                                                  ;00 AND 25200 MICRO SECONDS
7479
7480                                          ;NOW ONE SECTOR WRITE IS COMPLETE CHANGES WILL BE MADE
7481                                          ;FOR THE NEXT SECTOR THEN THE ABOVE WILL BE REPEATED
7482                                          ;TILL CYLINDER 256 IS REACHED
7483  026702  005237  026576                  INC     @#1$+10         ;CYLINDER HEADER DATA
7484  026706  005237  026616                  INC     @#2$+10         ;DATA
7485  026712  005237  026624                  INC     @#3$+4          ;CYLINDER COMMAND (RHCA)
7486  026716  005337  001172                  DEC     @#$TMP1         ;COUNT DOWN FOR 256
7487  026722  001321                          BNE     1$              ;BRANCH IF 256 NOT DONE
7488
7489
7490                                          ;NOW ALL 256 CYLINDERS HAVE CYLINDER NUMBER WRITTEN
7491                                          ;AS DATA ON SECTOR 0, TRACK 0, NOW RECALIBRATE FOLLOWED
7492                                          ;BY READ HEADER AND DATA THEN CHECK WILL BE DONE ON
7493                                          ;CYLINDER 0,1,2,4,8,16,32,64,128,256, AND 0
7494  026724  013737  026754  001110          MOV     @#4$,@#$LPERR   ;LOOP ON ERROR
7495  026732  005037  001172                  CLR     @#$TMP1         ;CYLINDER COUNTER
7496
7497                                          ;SETUP FOR CYLINDER 0
7498  026736  012737  010000  027036          MOV     #10000,@#5$+10  ;CYLINDER HEADER (DATA)
7499  026744  005037  027056                  CLR     @#6$+10         ;DATA
7500  026750  005037  027064                  CLR     @#7$+4          ;CYLINDER COMMAND (RHCA)
7501  026754                          4$:
7502
7503  026754  004737  030060                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7504                                                                  ;R3-RHDS1, R4-RHER1
7505                                                                  ;GIVE RH-11 INITIALIZE
```

```
7506                                                            ;SETUP UNIT NUMBER
7507
7508  026760  013777  004472  153172          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7509                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
7510                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7511                                                            ;'TIME' WILL ONLY SAVE
7512                                                            ;CURRENT CYLINDER ADDRESS
7513                                                            ;AND LOOK AHEAD REGISTERS
7514
7515
7516  026766  013746  002314          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
7517  026772  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7518                                                            ;ENABLE INTERRUPT
7519  026776  012677  153170          MOV     (SP)+,@RHCS1    ;GO WITH
7520                                                             ;6 IN RHCS1 FOR RECALIBRATE
7521                                                            ;WITH INTERRUPT ENABLED
7522
7523
7524  027002  104420                  WAT                     ;WAIT FOR DRY BIT TO SET
7525  027004  002214                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
7526  027006  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7527  027010  060650                  25000.                  ;ALLOW 250000 MICRO SECONDS
7528  027012  060650                  25000.                  ;DRY MUST SET BETWEEN
7529                                                            ;00 AND 500000 MICRO SECONDS
7530
7531                                  ;CLEAR READ INTO BUFFER WITH ALL ONES
7532  027014  004037  027730          JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
7533  027020  003420                  REINTO                  ;STARTING FROM REINTO
7534  027022  000404                  260.                    ;260. WORDS
7535  027024  177777                  -1                      ;FILL WITH -1
7536
7537
7538                                  ;FILL WRITE FROM BUFFER WITH EXPECTED HEADER
7539  027026                  5$:
7540
7541  027026  004037  027704          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
7542  027032  002354                  WRFROM                  ;LOCATION WHERE SAVED
7543  027034  000004                  4                       ;NUMBER OF WORDS SAVED
7544  027036  010000                  10000                   ;FIRST DATA WORD
7545  027040  000000                  0                       ;SECOND DATA WORD
7546  027042  000000                  0                       ;THIRD DATA WORD
7547  027044  000000                  0                       ;FOURTH DATA WORD
7548  027046                  6$:
7549  027046  004037  027730          JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
7550  027052  002364                  WRFROM+10                       ;STARTING FROM WRFROM+10
7551  027054  000400                  256.                    ;256. WORDS
7552  027056  000000                  0                       ;FILL WITH 0
7553
7554
7555                                  ;FILL READ HEADER AND DATA COMMAND
7556  027060                  7$:
7557
7558  027060  004037  032024          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7559  027064  000000                  0                       ;CYLINDER 0
7560  027066     000          .BYTE   0                       ;SECTOR 0
7561  027067     000          .BYTE   0                       ;TRACK 0
```

```
7562  027070  177374                          -256.-4          ;WORD COUNT (DATA)=256.+
7563                                                           ;4 HEADER WORDS
7564  027072  003420                          REINTO           ;BUS ADDRESS
7565                                                           ;STARTING ADDRESS OF DATA
7566                                                           ;BUFFER = REINTO
7567  027074  000000                          0                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7568  027076  014000                          ECI!FMT22                ;16 BITS PER WORD FORMAT
7569                                                           ;INHIBIT ECC CORRECTION
7570                                                           ;DO NOT INHIBIT HEADER COMPARE
7571  027100  002336                          REFOR            ;GET READY TO DO A REFOR
7572                                                           ;READ HEADER AND DATA WITH 72 IN RHCS1
7573
7574
7575  027102  004737  030140                  JSR     PC,@#CHECKT      ;CHECK DVA,RDY,MOL,DPR,DRY,VV
7576
7577
7578  027106  013777  004472  153044          MOV     @#RP4VEC,@RPVEC  ;SET RP04 VECTOR ADDRESS
7579                                                           ;TO 'TIME1' IF P-CLOCK IS PRESENT
7580                                                           ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7581                                                           ;'TIME' WILL ONLY SAVE
7582                                                           ;CURRENT CYLINDER ADDRESS
7583                                                           ;AND LOOK AHEAD REGISTERS
7584
7585                                          ;ONE SECTOR = 760 MICRO SECONDS, ONE REVOLUTION =
7586                                          ;16670 MICRO SECONDS, MAX SEEK = 52000 MICRO SECONDS
7587                                          ;MAX TIME = ONE REV + 1 SEEK + 1 SECTOR
7588
7589
7590  027114  013746  002336                  MOV     @#REFOR,-(SP)    ;GET READY TO MOVE COMMAND
7591  027120  052716  000101                  BIS     #GO!IE,(SP)      ;GET READY TO SET GO AND
7592                                                           ;ENABLE INTERRUPT
7593  027124  012677  153042                  MOV     (SP)+,@RHCS1     ;GO WITH
7594                                                             ;72 IN RHCS1 FOR READ DATA
7595                                                           ;WITH INTERRUPT ENABLED
7596
7597
7598  027130  104420                          WAT                      ;WAIT FOR RDY BIT TO SET
7599  027132  002172                          RHCS1                    ;WAIT FOR RHCS1 REGISTER
7600  027134  000200                          RDY                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7601  027136  006620                          3472.                    ;ALLOW 34720 MICRO SECONDS
7602  027140  006620                          3472.                    ;RDY MUST SET BETWEEN
7603                                                           ;00 AND 69440 MICRO SECONDS
7604
7605                                          ;CHECK READ WORDS AS ALL READ COMMANDS HAVE BEEN CHECKED
7606                                          ;DATA ERRORS MAY IMPLY "IMPLIED SEEK" ERRORS
7607
7608
7609  027142  004037  032070                  JSR     R0,@#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
7610  027146  002354                          WRFROM                   ;GOOD DATA      STARTS FROM WRFROM
7611  027150  003420                          REINTO                   ;TEST DATA STARTS FROM REINTO
7612  027152  000404                          260.                     ;260. WORDS TO BE COMPARED
7613  027154  027160                          8$                       ;RETURN TO 8$ ON ERROR
7614  027156  027164                          9$                       ;RETURN TO 9$ ON NO ERROR
7615
7616
7617  027160  104070                  8$:     ERROR   70               ;READ HEADER AND DATA ERROR
```

```
7618  027162  000207                          RTS     PC              ;DATA GIVES EXPECTED CYLINDER
7619
7620                                          ;NOW ONE SECTOR HAS BEEN CHECKED CHANGES WILL BE MADE
7621                                          ;TO READ THE NEXT SECTOR AND THE ABOVE PROGRAM WILL BE
7622                                          ;REPEATED
7623
7624  027164  005737  001172          9$:     TST     @#$TMP1         ;IS IT ZERO
7625  027170  001003                          BNE     10$             ;BRANCH IF NOT ZERO
7626  027172  005237  001172                  INC     @#$TMP1         ;ADD ONE
7627  027176  000407                          BR      11$             ;PUT ONE IN CYLINDER
7628  027200  022737  000400  001172  10$:    CMP     #256.,@#$TMP1   ;IS IT PASSED 256 CYLINDERS
7629  027206  101421                          BLOS    12$             ;YES SO GO TO ZERO
7630  027210  063737  001172  001172          ADD     @#$TMP1,@#$TMP1 ;DOUBLE
7631  027216  013737  001172  027056  11$:    MOV     @#$TMP1,@#6$+10 ;DATA
7632  027224  013746  001172                  MOV     @#$TMP1,-(SP)   ;GET CYLINDER NUMBER
7633  027230  052716  010000                  BIS     #FMT22,(SP)     ;INCLUDE FORMAT BIT
7634  027234  012637  027036                  MOV     (SP)+,@#5$+10   ;HEADER DATA (CYLINDER)
7635  027240  013737  001172  027064          MOV     @#$TMP1,@#7$+4  ;CYLINDER COMMAND (RHCA)
7636  027246  000137  026754                  JMP     @#4$            ;RETURN TO RECALIBRATE
7637  027252  022737  001000  001172  12$:    CMP     #512.,@#$TMP1   ;256 DONE
7638  027260  001414                          BEQ     13$             ;OUT
7639  027262  063737  001172  001172          ADD     @#$TMP1,@#$TMP1 ;DOUBLE
7640  027270  012737  010000  027036          MOV     #10000,@#5$+10  ;CYLINDER HEADER DATA
7641  027276  005037  027056                  CLR     @#6$+10         ;DATA
7642  027302  005037  027064                  CLR     @#7$+4          ;CYLINDER COMMAND (RHCA)
7643  027306  000137  026754                  JMP     @#4$            ;RETURN TO RECALIBRATE
7644
7645  027312                          13$:
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655                                          ;*****************************************************************
7656                                          ;*****************************************************************
7657                                          ;*TEST 36       END OF DRIVE
7658
7659                                          ;*      THIS IS THE END OF TEST FOR ONE DRIVE
7660                                          ;*      IF THERE ARE MORE DRIVES THEN THE PROGRAM
7661                                          ;*      JUMPS TO TEST 5 FOR NEXT DRIVE TEST
7662                                          ;*      END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE
7663
7664                                          ;*****************************************************************
7665  027312  000004                  TST36:  SCOPE
7666  027314  012767  000001  151662          MOV     #1,$TIMES       ;;DO 1 ITERATION
7667  027322  012767  000000  150446          MOV     #0,PS           ;REINSTATE PS TO 0
7668  027330  104400  027336                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
7669  027334  000425                          BR      64$             ;;GET OVER THE ASCIZ
7670                                          ;;.ASCIZ      <15><12>/TOTAL ERRORS ON THIS PASS ON  UNIT NO. /
7671  027410                          64$:
7672  027410  013746  004602                  MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
7673  027414  104410                          TYPDS
```

```
7674  027416  104400  027424              TYPE    ,.+4            ;;TYPE ASCIZ STRING
7675  027422  000402                      BR      65$             ;;GET OVER THE ASCIZ
7676                                      ;;.ASCIZ      /= /
7677  027430                      65$:
7678  027430  013746  001112              MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
7679  027434  104410                      TYPDS
7680  027436  005037  001112              CLR     @#$EPTTL        ;CLEAR TOTAL NUMBER OF ERRORS
7681  027442  005737  004612              TST     @#SELECT        ;STARTING FROM 200 ?
7682  027446  001415                      BEQ     3$              ;BRANCH IF YES
7683  027450  005067  151426              CLR     $TSTNM          ;CLEAR TEST NUMBER
7684  027454  005237  001100              INC     @#$PASS         ;INCREASE PASS COUNT
7685  027460  104400  027664              TYPE    ,$ENDMG         ;TYPE END PASS #
7686  027464  013746  001100              MOV     @#$PASS,-(SP)
7687  027470  104410                      TYPDS
7688  027472  104400  027701              TYPE    ,$ENULL
7689  027476  000137  007502              JMP     @#TST4          ;JUMP TEST 4
7690  027502  012737  177777  034666  3$: MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
7691  027510  005337  004604              DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENT
7692  027514  001413                      BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
7693  027516  013700  004602              MOV     @#UNIT,R0       ;UNIT UNDER TEST
7694  027522  012701  004562              MOV     #UNITS,R1       ;TABLE
7695  027526  022100                  1$: CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
7696  027530  001401                      BEQ     2$              ;BRANCH IF YES
7697  027532  000775                      BR      1$              ;BRANCH IF NO
7698  027534  011137  004602          2$: MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
7699  027540  000137  007502              JMP     @#TST4          ;GO FOR NEXT TESTS.
```

```
 7700                                ;*****************************************************************
 7701
 7702                                .SBTTL  END OF PASS ROUTINE
 7703
 7704                                ;*INCREMENT THE PASS NUMBER ($PASS)
 7705                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
 7706                                ;*IF THERES A MONITOR GO TO IT
 7707                                ;*IF THERE ISN'T JUMP TO TST1
 7708
 7709  027544                        $EOP:
 7710  027544  000004                       SCOPE
 7711  027546  005067  151330               CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 7712  027552  005067  151426               CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 7713  027556  005267  151316               INC     $PASS           ;;INCREMENT THE PASS NUMBER
 7714  027562  042767  100000  151310       BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 7715  027570  005327                       DEC     (PC)+           ;;LOOP?
 7716  027572  000001                $EOPCT: .WORD   1
 7717  027574  003031                       BGT     $DOAGN          ;;YES
 7718  027576  012737                       MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 7719  027600  000001                $ENDCT: .WORD   1
 7720  027602  027572                       $EOPCT
 7721  027604  104400  027664               TYPE    ,$ENDMG         ;;TYPE "END PASS #"
 7722  027610  016746  151264               MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 7723  027614  104410                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 7724  027616  104400  027701               TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 7725  027622  013700  000042        $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 7726  027626  001414                       BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 7727  027630  022700  027650               CMP     #$ENDAD,R0      ;;IS MONITOR ACT11?
 7728  027634  001004                       BNE     $RESET          ;;NO--BRANCH (IT'S XXDP)
 7729  027636  022760  177777  000002       CMP     @-1,2(R0)       ;;YES--IS THIS THE LAST PASS?
 7730  027644  001001                       BNE     $ENDAD          ;;NO--MAKE ANOTHER PASS
 7731  027646  000005                $RESET: RESET                   ;;CLEAR THE WORLD
 7732  027650  004710                $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 7733  027652  000240                       NOP                     ;;SAVE ROOM
 7734  027654  000240                       NOP                     ;;FOR
 7735  027656  000240                       NOP                     ;;ACT11
 7736  027660  000137  006100        $DOAGN: JMP     @#TST1          ;;RETURN
 7737  027664  005015  047105  020104 $ENDMG: .ASCIZ  <15><12>/END PASS #/
 7738  027672  040520  051523  021440
 7739  027700     000
 7740  027701     377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 7741
 7742
 7743
 7744                                       .SBTTL  SUBROUTINES
 7745
 7746
 7747
 7748
 7749
 7750
 7751
 7752                                ;THIS FILLS MEMORY WITH GIVEN DATA
 7753                                ;USED CHIEFLY FOR HEADER INFORMATION
 7754                                ;CALL IS
 7755                                ;       JSR     R0,@#FLHEAD     ;FILL HEADER
```

```
                                ;       LOC                     ;LOCATION WHERE SAVED
                                ;       XN                      ;NUMBER OF WORDS
                                ;       XD1                     ;DATA REPEATED XN TIMES
                                ;       XD2                     ;DATA REPEATED XN TIMES
                                ;
                                ;
                                ;
027704                          FLHEAD:
027704  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
027706  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
027710  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF WHERE TO SAVE
027712  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS
                                        ;NOW FILL DATA
027714  012021                  1$:     MOV     (R0)+,(R1)+     ;SAVE DATA
027716  005302                          DEC     R2              ;DECREMENT COUNT
027720  001375                          BNE     1$              ;BRANCH IF INCOMPLETE
027722  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
027724  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
027726  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
027730                          CLAREA:
027730  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
027732  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
027734  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
027736  012001                          MOV     (R0)+,R1        ;FROM
027740  012002                          MOV     (R0)+,R2        ;NUMBER
027742  012003                          MOV     (R0)+,R3        ;DATA
027744  010321                  1$:     MOV     R3,(R1)+        ;MOVE DATA
027746  005302                          DEC     R2              ;COUNT
027750  001375                          BNE     1$              ;BRANCH IF NOT COMPLETE
027752  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
027754  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
027756  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
027760  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
7812
7813                                    ;THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
7814                                    ;WITH GIVEN VALUE
7815                                    ;CALL IS
7816                                    ;       JSR     R0,@#FILLRE
7817                                    ;       RHXX            ;REGISTER NAME
7818                                    ;       D               ;DATA
7819                                    ;
7820
7821  027762                            FILLRE:
7822  027762  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7823  027764  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
7824  027766  012001                            MOV     (R0)+,R1        ;ADDRESS OF ADDRESS OF REGISTER
7825  027770  012002                            MOV     (R0)+,R2        ;DATA
7826  027772  162701  002164                   SUB     #RHWC,R1        ;OFSET
7827  027776  010261  004476                   MOV     R2,SAVERE(R1)   ;DATA IS MOVED IN
7828  030002  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
7829  030004  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
7830  030006  000200                            RTS     R0              ;RETURN TO MAIN PROGRAM
7831
7832
7833
```

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

030010  012077  152170          SRCH:   MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
030014  012077  152156          SRCH:   MOV     (R0)+,@RHDST    ;SET DESIRED SECTOR/TRACK ADDRESS
030020  013777  002322  152144          MOV     @#SERCH,@RHCS1  ;GET READY FOR SEARCH
                                                                ;WITH 30 IN RHCS1
030026  000200                          RTS     R0


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

030030  012077  152150          SEEKCY: MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
030034  013777  002340  152130          MOV     @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
030042  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
7862
7863                                    ;THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
7864                                    ;CALL IS
7865                                    ;       JSR     R0,@#OFSET
7866                                    ;       0               ;MICRO INCHES OFSET
7867
7868  030044  052077  152132  OFSET:    BIS     (R0)+,@RHOF     ;SET OFSET REGISTER
7869  030050  013777  002342  152114    MOV     @#OFSETC,@RHCS1 ;MOV14 INTO RHCS1
7870  030056  000200                    RTS     R0              ;RETURN TO MAIN PROGRAM
7871
7872
7873  030060  013701  002172  CLDISK:   MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
7874  030064  013702  002170            MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
7875  030070  013703  002214            MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
7876  030074  013704  002174            MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1
7877
7878  030100  012712  000040            MOV     #CLR,@R2                ;CLEAR ALL REG.
7879  030104  013712  004602            MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
7880  030110  005011                    CLR     @R1                     ;CLEAR FUNCTION BITS
7881  030112  000207                    RTS     PC
```

```
7882
7883
7884
7885
7886                                    ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
7887                                    ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
7888  030114  000000                    PCJSR:  0                       ;PC OF JSR
7889
7890
7891
7892
7893  030116  011637  030114            CHECK:  MOV     (SP),@#PCJSR            ;SAVE PC OF JSP+4
7894  030122  162737  000004  030114            SUB     #4,@#PCJSR      ;GET PC OF JSR
7895  030130  011346                    MOV     @R3,-(SP)       ;GET RHDS1
7896  030132  052716  000100                    BIS     #VV,(SP)        ;DONT CHECK VV BIT
7897  030136  000406                    BR      CHECKC          ;GOTO COMMON CHECK ROUTINE
7898  030140  011637  030114            CHECKT: MOV     (SP),@#PCJSR            ;SAVE PC OF JSR+4
7899  030144  162737  000004  030114            SUB     #4,@#PCJSR      ;GET PC OF JSR
7900  030152  011346                    MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
7901  030154  011146                    CHECKC: MOV     @R1,-(SP)       ;GET CS1
7902  030156  042716  173577                    BIC     #173577,(SP)            ;CLEAR UNWANTED BITS
7903  030162  022726  004200                    CMP     #DVA!RDY,(SP)+          ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
7904                                                                    ;AND BE READY
7905  030166  001403                    BEQ     3$                      ;BRANCH IF GOOD
7906  030170  011137  001122                    MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
7907  030174  104062                    ERROR   62                      ;RHCS1 DID NOT HAVE DEVICE
7908                                                                    ;AVAILABLE RIGHT AT THE START
7909                                                                    ;ALL OTHER BITS SHOULD BE 0
7910  030176  042716  102000            3$:     BIC     #ATA!LBT,(SP)           ;CLEAR UNWANTED BITS
7911  030202  022726  010700                    CMP     #MOL!DPR!DRY!VV,(SP)+;RHDS1 SHOULD HAVE THESE SET
7912  030206  001403                    BEQ     7$                      ;BRANCH IF GOOD
7913  030210  011337  001122                    MOV     @R3,@#$BDADR    ;BAD DATA IN  REGISTER (RHDS1)
7914  030214  104061                    ERROR   61                      ;RHDS1 HAS SOME BITS OTHER
7915                                                                    ;THAN MOL, DRY, DPR,VV SET
7916                                                                    ;ALL OTHER BITS SHOULD BE 0
7917  030216  000207                    7$:     RTS     PC                      ;RETURN TO TEST NO.
7918
```

```
                                        ;THIS IS A SUBROUTINE TO SAVE REGISTERS
                                        ;IN THE REGISTER TABLE TO ANY LOCATION
                                        ;THE CALL IS
                                        ;JSR     R0,@#SAVER
                                ;       F       ;FROM
                                ;       T       ;TO
                                ;       N       ;NUMBER OF WORDS SAVED
                                ;F MUST ALWAYS BE RHCS1
                                ;T MUST ALWAYS BE SAVRE

030220                          SAVER:
030220  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030222  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
030224  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
030226  012001                          MOV     (R0)+,R1        ;FROM
030230  012002                          MOV     (R0)+,R2        ;TO
030232  012003                          MOV     (R0)+,R3        ;NUMBER
030234  013122                  1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
030236  005303                          DEC     R3              ;COUNT
030240  001375                          BNE     1$              ;BRANCH IF NOT DONE
030242  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
030244  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
030246  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
030250  000200                          RTS     R0


                                        ;WHEN AN EVENT IS TO BE TIMED THE RP04 VECTORS TO "TIME 1"
                                        ;PRIORITY OF PROCESS OR IS 4
                                        ;PRIORITY OF TRAPS MUST BE 6
                                        ;PRIORITY OF RP04 INTERRUPTS IS 7
                                        ;

030252  005077  151754        TIME1:  CLR     @PCLCSR                 ;STOP THE CLOCK
030256  017737  151754  030310          MOV     @PCLCTR,@#WAITTM ;GET TIME ON CLOCK
030264  017737  151736  004550  TIME2:  MOV     @RHCC,@#FINACC          ;GET CURRENT CYLINDER
030272  017737  151732  004546          MOV     @RHLA,@#FINALA          ;GET LOOK AHEAD
030300  000002                          RTI                     ;RETURN TO WAIT P OR WAIT.T


                                        ;THIS IS A WAIT LOOP WHEN AN EVENT IS TO BE TIMED
                                        ;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

030302  000000                  WAITPC: 0                       ;WAT PC
030304  000000                  WAITRE: 0                       ;WAIT ON REGISTER ADDRESS
030306  000000                  WAITBT: 0                       ;WAIT ON BIT
030310  000000                  WAITTM: 0                       ;WAITED TIME
030312  005037  030310          WAIT.P: CLR     @#WAITTM        ;CLEAR WAITED TIME
030316  005077  151712                  CLR     @PCLBUF ;CLEAR COUNT SET BUFFER
030322  012777  000021  151702         MOV     #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
030330  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
030332  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030334  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
030336  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
030340  016600  000010                  MOV     10(SP),R0       ;R0 HAS ADDRESS OF NEXT LOCATION
030344  010037  030302                  MOV     R0,@#WAITPC     ;NOW WAITPC HAS WAT PC + 2
030350  162737  000002  030302          SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
030356  013037  030304                  MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
030362  012037  030306                  MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
030366  012001                          MOV     (R0)+,R1        ;R1 HAS TIME IN 10 MSEC
030370  012002                          MOV     (R0)+,R2        ;R2 HAS TOLERANCE IN 10 MSEC
030372  010066  000010                  MOV     R0,10(SP)       ;RESTORE RETURN ON STACK

                                        ;THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
030376  013703  030550                  MOV     @#TIMCNT,R3     ;R3 IS A TEMPORARY COUNTER
030402  033777  030306  177674  1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
030410  001025                          BNE     4$              ;BRANCH IF YES
030412  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
030414  001372                          BNE     1$
030416  013703  030550                  MOV     @#TIMCNT,R3     ;TEMPORARY COUNTER
030422  033777  030306  177654  2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
030430  001015                          BNE     4$              ;BRANCH IF YES
030432  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
030434  001372                          BNE     2$
030436  017737  177642  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
030444  032777  000100  151520          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
030452  001402                          BEQ     3$              ;BRANCH IF YES
030454  104001                          ERROR   1               ;RP04 DID NOT INTERRUPT
030456  000427                          BR      7$              ;OUT
030460  104002                  3$:     ERROR   2               ;RP04 INTERRUPTED BUT WAITED
                                        ;ON BIT DID NOT OCCUR
                                        ;EVEN AFTER TWO COUNT DOWNS
                                        ;FROM 177777 TO 0
030462  000425                          BR      7$              ;OUT

                                        ;NOW TIME AND TOLERANCE WILL BE CHECKED
030464  017737  177614  001126  4$:     MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
030472  032777  000100  151472          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
```

```
8031  030500  001402                        BEQ     5$              ;BRANCH IF YES
8032  030502  104003                        ERROR   3               ;INTERRUPT DID NOT OCCUR EVEN
8033                                                                  ;AFTER ONE BNE AND ONE MOV
8034                                                                  ;OF THE WAITED ON BIT SETTING
8035  030504  000414                        BR      7$              ;OUT
8036  030506  160201                5$:     SUB     R2,R1           ;R1 NOW HAS LOWER LIMIT OF TIME
8037  030510  023701  030310                CMP     @#WAITTM,R1     ;FOR GOOD RESULTS, WAITTM
8038                                                                  ;MUST BE GREATER OR EQUAL
8039                                                                  ;TOR1
8040  030514  103002                        BHIS    6$              ;BRANCH IF GOOD
8041  030516  104004                        ERROR   4               ;BIT DID OCCUR BUT TIME
8042                                                                  ;TAKEN IS BELOW LOWER LIMIT
8043  030520  000406                        BR      7$              ;OUT
8044
8045  030522  060202                6$:     ADD     R2,R2           ;DOUBLE TOLERANCE
8046  030524  060201                        ADD     R2,R1           ;R1 NOW HAS UPPER LIMIT OF TIME
8047  030526  020137  030310                CMP     R1,@#WAITTM     ;FOR GOOD RESULTS, WAITTM
8048                                                                  ;MUST BE LESS OR EQUAL TO R1
8049  030532  103001                        BHIS    7$              ;BRANCH IF GOOD
8050  030534  104004                        ERROR   4               ;BIT DID OCCUR BUT TIME TAKEN
8051                                                                  ;IS ABOVE UPPER LIMIT
8052  030536                        7$:
8053  030536  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8054  030540  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8055  030542  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8056  030544  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8057  030546  000002                        RTI                     ;RETURN TO MAIN TEST
8058
8059
8060
8061
8062
8063
8064                                ;THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
8065                                ;NO TIMING IS DONE
8066                                ;CALL IS
8067                                ;       WAT
8068                                ;       A       ;ABSOLUTE REGISTER ADDRESS
8069                                ;       B       ;BIT WAITE) FOR
8070                                ;       TA      ;TIME-NOT USED HERE
8071                                ;       TO      ;TIME-NOT USED HERE
8072                                ;P3-IS A TEMPORARY COUNTER
8073  030550  177777                TIMCNT: 177777                  ;COUNT FOR WAIT LOOP
8074
8075  030552                        WAIT.T:
8076  030552  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8077  030554  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8078  030556  016600  000004                MOV     4(SP),R0        ;R0 HAS ADDRESS OF NEXT LOCATION
8079  030562  010037  030302                MOV     R0,@#WAITPC     ;WAT PC +2 IS IN WAITPC
8080  030566  162737  000002  030302        SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
8081  030574  013037  030304                MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
8082  030600  012037  030306                MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
8083  030604  022020                        CMP     (R0)+,(R0)+     ;DUMP NEXT TWO WORDS-TA, TO
8084  030606  010066  000004                MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
8085
8086                                        ;THIS HAS THE TWO COUNT DOWNS FROM 177777
```

```
8087  030612  013703  030550                MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
8088  030616  033777  030306  177460  1$:   BIT     @#WAITBT,@WAITPE ;IS REQUIRED BIT THERE
8089  030624  001025                        BNE     4$              ;BRANCH IF YES
8090  030626  005303                        DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
8091  030630  001372                        BNE     1$
8092  030632  013703  030550                MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
8093  030636  033777  030306  177440  2$:   BIT     @#WAITBT,@WAITPE ;IS REQUIRED BIT THERE
8094  030644  001015                        BNE     4$              ;BRANCH IF YES
8095  030646  005303                        DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
8096  030650  001372                        BNE     2$
8097  030652  017737  177426  001126        MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
8098  030660  032777  000100  151304        BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
8099  030666  001402                        BEQ     3$              ;BRANCH IF YES
8100  030670  104001                        ERROR   1               ;RP04 DID NOT INTERRUPT
8101                                                                ;BIT DID NOT OCCUR
8102  030672  000414                        BR      5$              ;OUT
8103  030674  104002                3$:     ERROR   2               ;RP04 INTERRUPTED BUT
8104                                                                ;WAITED ON BIT DID NOT OCCUR
8105                                                                ;EVEN AFTER TWO COUNT DOWNS
8106                                                                ;FROM 177777 TO 0
8107  030676  000412                        BR      5$              ;OUT
8108
8109                                        ;BIT DID SET SO CHECK IF INTERRUPT OCCURED
8110  030700  011010                4$:     MOV     (0),(0)         ;ALLOW TIME FOR INTERRUPT
8111  030702  032777  000100  151262        BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
8112  030710  001405                        BEQ     5$              ;BRANCH IF YES
8113  030712  017737  177366  001126        MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
8114  030720  104003                        ERROR   3               ;INTERRUPT DID NOT OCCUR
8115                                                                ;EVEN AFTER ONE BNE OF
8116                                                                ;THE WAITED ON BIT OCCURING
8117  030722  000400                        BR      5$              ;OUT
8118  030724                        5$:
8119  030724  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8120  030726  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8121  030730  000002                        RTI                     ;RETURN TO MAIN TEST
```

```
                                ;THIS CHANGES REGISTER SAVED VALUE
                                ;CALL IS
                                ;       JSR     R0,@#CHREG
                                ;       P                       ;REGISTER TO BE CHANGED
                                ;       N                       ;NUMBER OF BITS TO BE CHANGED
                                ;       NEW                     ;NEW VALUE OF BIT MUST BE 0 OR1
                                ;       P                       ;POSITION OF BIT TO BE CHANGED
                                ;NEW AND P WILL BE REPEATED N NUMBER OF TIMES
030732                          CHREG:
030732  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030734  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
030736  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
030740  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF CHANGES
030742  162701  002164                  SUB     #RHWC,R1        ;R1 HAS OFSET OF REQUIRED REGISTER
030746  005720                  1$:     TST     (R0)+           ;IS A BIC OR A BIS TO BE DONE
030750  001403                          BEQ     2$              ;BRANCH IF A BIC IS REQUIRED
030752  052061  004476                  BIS     (R0)+,SAVERE(R1) ;SET REQUIRED BIT
030756  000402                          BR      3$              ;BRANCH TO DECREMENT COUNT
030760  042061  004476          2$:     BIC     (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
030764  005302                  3$:     DEC     R2              ;DECREMENT NUMBER OF CHANGES
030766  001367                          BNE     1$              ;BRANCH IF NOT COMPLETE
030770  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
030772  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
030774  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM


                                ;THIS FILLS A BLOCK WITH INCREMENTAL DATA
                                ;CALL IS
                                ;       JSR     R0,@#FILL
                                ;       F                       ;FROM
                                ;       N                       ;NUMBER OF WORDS
                                ;       S                       ;STARTING VALUE OF DATA
                                ;       I                       ;INCREMENT DATA BY

030776                          FILL:
030776  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
031000  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
031002  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
031004  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
031006  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS WHERE DATA IS TO GO
031010  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS TO BE FILLED
031012  012003                          MOV     (R0)+,R3        ;STARTING VALUE OF DATA
031014  012004                          MOV     (R0)+,R4        ;R4 HAS INCREMENT
                                ;NOW DATA WILL BE FILLED
031016  010321                  1$:     MOV     R3,(R1)+        ;FILL DATA
031020  060403                          ADD     R4,R3           ;GET NEXT VALUE OF DATA
031022  005302                          DEC     R2              ;DECREMENT COUNT
031024  001374                          BNE     1$              ;BRANCH IF ALL NOT DONE
031026  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
031030  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
031032  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
031034  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
8178  031036  000200                        RTS     R0              ;RETURN TO MAIN PROGRAM
8179
8180
8181
8182
8183
8184
8185                                ;THIS IS A SUBROUTINE TO COMPARE REGISTERS
8186                                ;GOOD DATA IS ALREADY SAVED IN 'SAVERE'
8187                                ;TEST DATA IS IN THE REGISTERS
8188                                ;CALL IS
8189                                ;       JSP     R0,@#COMREG
8190                                ;       SAVERE                  ;GOOD DATA
8191                                ;       RHCS1                   ;ADDRESS OF ADDRESS TEST DATA
8192                                ;       N.                      ;RETURN FOR ERROR
8193                                ;       RG                      ;RETURN FOR GOOD COMPARISON
8194                                ;ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
8195                                ;'REGADR' HAS REGISTER ADDRESS
8196
8197  031040                        COMREG:
8198  031040  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8199  031042  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8200  031044  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8201  031046  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8202  031050  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8203  031052  012001                        MOV     (R0)+,R1        ;R1 HAS ADDRESS OF GOOD DATA
8204  031054  012002                        MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
8205  031056  012003                        MOV     (R0)+,R3        ;R3 HAS NUMBER OF WORDS
8206  031060  012004                        MOV     (R0)+,R4        ;R4 HAS RETURN FOR ERROR
8207  031062  011000                        MOV     (R0),R0         ;R0 HAS RETURN ON NO ERROR
8208                                ;NOW SAVE REGISTERS
8209  031064  004737  031764               JSR     PC,@#PUTREG     ;SAVE REGISTERS
8210  031070  113737  004523  002267        MOVB    @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF RHAS SAME
8211  031076  012705  177776               MOV     #-2,R5          ;PRESET R5 TO -2
8212                                ;NOW COMPARES WILL MADE
8213  031102  062705  000002        1$:    ADD     #2,R5           ;INCREMENT TO INDEX
8214  031106  022122                        CMP     (R1)+,(R2)+     ;COMPARE REGISTER CONTENTS
8215  031110  001420                        BEQ     2$              ;BRANCH IF GOOD
8216  031112  014137  001124               MOV     -(R1),@#$GDDAT  ;SAVE GOOD DATA
8217  031116  014237  001126               MOV     -(R2),@#$BDDAT  ;SAVE BAD DATA
8218  031122  016537  002164  004464        MOV     RHWC(R5),@#REGADR       ;SAVE ADDRESS OF FAILING REGISTER
8219  031130  004714                        JSR     PC,@R4          ;RETURN TO MAIN PROGRAM
8220                                                                ;TO PRINT ERROR
8221  031132  022122                        CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
8222  031134  013746  177570               MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
8223  031140  042716  177177               BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
8224  031144  022726  000200               CMP     #SW07,(SP)+     ;IS 7 SET AND 8 DOWN
8225  031150  001402                        BEQ     3$              ;BRANCH OUT IF YES
8226  031152  005303                2$:     DEC     R3              ;ARE ALL COMPARES DONE
8227  031154  001352                        BNE     1$              ;BRANCH IF NOT COMPLETE
8228
8229  031156                        3$:
8230  031156  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
8231  031160  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
8232  031162  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8233  031164  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
  8234  031166  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
  8235  031170  000200                        RTS     R0              ;RETURN TO MAIN PROGRAM
  8236  031172  000000              48:       .WORD   0               ;TEMP STORAGE
```

```
8237
8238
8239                                    ;HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
8240                                    ;ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
8241                                    ;PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
8242
8243                                    ;WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
8244                                    ;THE PROGRAM GOES BACK TO CAN BE CHANGED.
8245                                    ;THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
8246                                    ;1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
8247                                    ;2.  LOOP ON ERROR SWITCH MUST BE SET
8248                                    ;3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
8249                                    ;IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
8250                                    ;THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
8251                                    ;TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
8252                                    ;THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
8253                                    ;COMES TO THE END OF THE TEST UNDER CONSIDERATION.
8254                                    ;
8255                                    ;AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
8256                                    ;NORMAL OPERATION WILL CONTINUE.
8257
8258  031174  000000                   TESTAD: 0                       ;FIRST ADDRESS OF TEST
8259  031176                          OPERSEL:
8260  031176  005067  146574                   CLR     PS              ;MAKE PROCESSOR STATUS ZERO
8261  031202  012737  177777  034666           MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
8262  031210  104400  031216                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
8263  031214  000421                           BR      64$             ;;GET OVER THE ASCIZ
8264                                          ;;.ASCIZ      <15><12>/THE PROGRAM WAS IN TEST NUMBER /
8265  031260                          64$:
8266  031260  013746  004470                   MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
8267  031264  104402                           TYPOC                   ;NUMBER
8268  031266  104400  031274                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
8269  031272  000414                           BR      65$             ;;GET OVER THE ASCIZ
8270                                          ;;.ASCIZ      <15><12>/THE LOOP BACK PC WAS /
8271  031324                          65$:
8272  031324  013746  001110                   MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
8273  031330  104402                           TYPOC
8274  031332  104400  001215                   TYPE    ,$CRLF
8275  031336  104400  031344                   TYPE    .,.+4           ;;TYPE ASCIZ STRING
8276  031342  000430                           BR      66$             ;;GET OVER THE ASCIZ
8277                                          ;;.ASCIZ      <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
8278  031424                          66$:
8279  031424  104400  031432                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
8280  031430  000420                           BR      67$             ;;GET OVER THE ASCIZ
8281                                          ;;.ASCIZ      <15><12>/TYPE THE FIRST PC OF THE TEST/
8282  031472                          67$:
8283  031472  104400  031500                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
8284  031476  000432                           BR      68$             ;;GET OVER THE ASCIZ
8285                                          ;;.ASCIZ      <15><12>/TO BE LOOPED ON FOLLOWED BY A CARRIAGE RETURN /<15><12>
8286  031564                          68$:
8287  031564  104416                           RDOCT
8288  031566  062716  000002                   ADD     #2,(SP)         ;GET LPADR
8289  031572  012637  001106                   MOV     (SP)+,@#$LPADR
8290  031576  104400  031604                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
8291  031602  000417                           BR      69$             ;;GET OVER THE ASCIZ
8292                                          ;;.ASCIZ      <15><12>/TYPE THE PC WHERE YOU WANT/
```

```
 8293  031642                          69$:
 8294  031642  104000  031650                TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8295  031646  000440                      BR      70$             ;;GET OVER THE ASCIZ
 8296                                      ;;.ASCIZ      <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETUR
 8297  031750                          70$:
 8298  031750  104416                      RDOCT
 8299  031752  012537  001110              MOV     (SP)+,@#$LPERR  ;GET LPERR
 8300  031756  013746  001106              MOV     @#$LPADR,-(SP)
 8301  031762  000002                      RTI
 8302
 8303
 8304
 8305
 8306
 8307
 8308                                      ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
 8309                                      ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "FC2"
 8310                                      ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
 8311                                      ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
 8312                                      ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
 8313
 8314
 8315
 8316
 8317  031764                          PUTREG:
 8318  031764  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8319  031766  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8320  031770  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8321  031772  012700  002164              MOV     #PHWC,R0        ;STARTING ADDRESS OF REG
 8322  031776  012701  002242              MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
 8323  032002  012702  000022              MOV     #PHCC-PHWC+2/2,R2 ;NUMBER OF REG. INTO R2
 8324  032006  013021                  10$: MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
 8325  032010  005302                      DEC     R2
 8326  032012  001375                      BNE     10$
 8327  032014  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
 8328  032016  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
 8329  032020  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
 8330  032022  000207                      RTS     PC
```

```
8331                                    ;THIS IS A DATA  COMMAND SETUP SUBROUTINE
8332                                    ;THE CALL IS
8333                                    ;       JSR     R0,@#RUN
8334                                    ;       C                       ;CYLINDER
8335                                    ;.BYTE  S                       ;SECTOR
8336                                    ;.BYTE  T                       ;TRACK
8337                                    ;       -W                      ;WORD COUNT
8338                                    ;       B                       ;BUS ADDRESS
8339                                    ;       BAI                     ;BUS ADDRESS INHIBIT
8340                                    ;       FMT22!ECI!HCI           ;FMT22=1 =16 BIT WORDS
8341                                    ;                               ;ECI = ECC CORRECTION INHIBIT
8342                                    ;                               ;HCI = HEADER COMPARE INHIBIT
8343                                    ;       COM                     ;COMMAND ADDRESS
8344  032024  012077  150154           RUN:    MOV     (R0)+,@RHCA     ;CYLINDER
8345  032030  012077  150142                   MOV     (R0)+,@RHDST    ;DESIRED SECTOR/TRACK
8346  032034  012077  150124                   MOV     (R0)+,@RHWC     ;WORD COUNT
8347  032040  012077  150122                   MOV     (R0)+,@RHBA     ;BUS ADDRESS
8348  032044  013746  004602                   MOV     @#UNIT,-(SP)    ;GET UNIT NO
8349  032050  052016                           BIS     (R0)+,(SP)      ;SET BUS ADDRESS INHIBIT
8350  032052  012677  150112                   MOV     (SP)+,@RHCS2    ;UNIT NO AND BAI TO RHCS2
8351  032056  012077  150120                   MOV     (R0)+,@RHOF     ;FORMAT, ECC INHIBIT, HEADER
8352                                                                    ;COMPARE, IF THERE
8353  032062  013077  150104                   MOV     @(R0)+,@RHCS1   ;COMMAND IN RHCS1
8354  032066  000200                           RTS     R0              ;RETURN TO MAIN PROGRAM
8355
8356
8357
8358
8359                                    ;THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
8360                                    ;R1 HAS GOOD DATA BUFFER ADDRESS
8361                                    ;R2 HAS TEST DATA BUFFER ADDRESS
8362                                    ;R5 HAS ADDRESS OF RETURN ON ERROR
8363                                    ;R3 HAS NUMBER OF WORDS TO BE COMPARED
8364                                    ;R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
8365                                    ;CALL IS
8366                                    ;       JSR     R0,@#COMPAR
8367                                    ;       G                       ;ADDRESS OF GOOD DATA
8368                                    ;       T                       ;ADDRESS OF TEST DATA
8369                                    ;       N                       ;NUMBER OF WORDS TO BE COMPARED
8370                                    ;       RE                      ;RETURN ON ERROR
8371                                    ;       RG                      ;RETURN ON NO ERROR
8372
8373
8374
8375  032070                            COMPAR:
8376  032070  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8377  032072  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8378  032074  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8379  032076  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8380  032100  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8381  032102  012001                           MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
8382  032104  012002                           MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
8383  032106  012003                           MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
8384  032110  012005                           MOV     (R0)+,R5        ;RETURN ON ERROR
8385  032112  011000                           MOV     (R0),R0         ;RETURN ON NO ERROR
8386  032114  010304                           MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
```

```
 8387  032116  005204                        INC     R4
 8388  032120  010437  004466          1$:   MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
 8389  032124  022122                        CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
 8390  032126  001417                        BEQ     2$              ;BRANCH IF GOOD
 8391
 8392  032130  014137  001124                MOV     -(R1),@#$GDDAT  ;GOOD DATA
 8393  032134  014237  001126                MOV     -(R2),@#$BDDAT  ;BAD DATA
 8394  032140  160337  004466                SUB     R3,@#ERWORD     ;ERROR WORD NO.
 8395  032144  004715                        JSR     PC,@R5  ;RETURN TO PRINT ERROR
 8396  032146  022122                        CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
 8397  032150  013746  177570                MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
 8398  032154  042716  177177                BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
 8399  032160  022726  000200                CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
 8400  032164  001402                        BEQ     3$              ;BRANCH OUT IF YES
 8401  032166  005303                2$:     DEC     R3              ;COUNT
 8402  032170  001353                        BNE     1$              ;BRANCH IF ALL NOT DEVICE
 8403  032172                        3$:
 8404  032172  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 8405  032174  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 8406  032176  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 8407  032200  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 8408  032202  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 8409  032204  000200                        RTS     R0              ;RETURN TO MAIN PROGRAM
 8410                                ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
 8411                                ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
 8412
 8413  032206                        BASECH:
 8414  032206  104400  032214                TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8415  032212  000424                        BR      64$             ;;GET OVER THE ASCIZ
 8416                                        ;;.ASCIZ        <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS /
 8417  032264                        64$:
 8418  032264  013746  002172                MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
 8419  032270  104402                        TYPOC
 8420  032272  104400  032300                TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8421  032276  000425                        BR      65$             ;;GET OVER THE ASCIZ
 8422                                        ;;.ASCIZ        <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
 8423  032352                        65$:
 8424  032352  104416                        RDOCT
 8425  032354  012700  002162                MOV     #RHDB,R0        ;GET STARTING ADDRESS OF RGISTERS
 8426  032360  012701  000024                MOV     #20.,R1         ;NUMBER OF REGISTERS
 8427  032364  042710  177700          1$:   BIC     #^C77,(R0)      ;CLEAR OLD BASE
 8428  032370  051620                        BIS     (SP),(R0)+      ;SET NEW BASE
 8429  032372  005301                        DEC     R1              ;COUNT
 8430  032374  001373                        BNE     1$              ;BRANCH IF 20 NOT DONE
 8431  032376  104400  032404                TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8432  032402  000417                        BR      66$             ;;GET OVER THE ASCIZ
 8433                                        ;;.ASCIZ        <15><12>/PRESENT VECTOR ADDRESS IS /
 8434  032442                        66$:
 8435  032442  013746  002160                MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
 8436  032446  104402                        TYPOC
 8437  032450  104400  032456                TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8438  032454  000437                        BR      67$             ;;GET OVER THE ASCIZ
 8439                                        ;;.ASCIZ        <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY 'CR
 8440  032554                        67$:
 8441  032554  104416                        RDOCT
 8442  032556  012637  002160                MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
```

```
 8443  032562  104400  032570                TYPE    .+4             ;;TYPE ASCIZ STRING
 8444  032566  000421                        BR      68$             ;;GET OVER THE ASCIZ
 8445                                        ;;.ASCIZ        <15><12>/RESTART PROGRAM FROM 200 OR 210/
 8446  032632                          68$:
 8447  032632  104400  032640                TYPE    .+4             ;;TYPE ASCIZ STRING
 8448  032636  000414                        BR      69$             ;;GET OVER THE ASCIZ
 8449                                        ;;.ASCIZ        <15><12>/NEW BASE WILL REMAIN/
 8450  032670                          69$:
 8451  032670  013746  002172                MOV     @#RHCS1,-(SP)
 8452  032674  104402                        TYPOC
 8453  032676  104400  032704                TYPE    .+4             ;;TYPE ASCIZ STRING
 8454  032702  000415                        BR      70$             ;;GET OVER THE ASCIZ
 8455                                        ;;.ASCIZ        <15><12>/NEW VECTOR WILL REMAIN /
 8456  032736                          70$:
 8457  032736  013746  002160                MOV     @#RPVEC,-(SP)
 8458  032742  104402                        TYPOC
 8459  032744  104400  032752                TYPE    .+4             ;;TYPE ASCIZ STRING
 8460  032750  000416                        BR      71$             ;;GET OVER THE ASCIZ
 8461                                        ;;.ASCIZ        <15><12>/UNTIL PROGRAM IS RELOADED/
 8462  033006                          71$:
 8463  033006  000000                        HALT
 8464
```

```
 8465  033010                          PPVECT:
 8466  033010  104403  033016                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 8467  033014  000411                          BR      64$             ;;GET OVER THE ASCIZ
 8468                                          ;;.ASCIZ      /TRAPED FROM PC = /
 8469  033040                          64$:
 8470  033040  104402                          TYPOC                           ;TYPE FROM PC
 8471  033042  012777  033010  147110          MOV     #PPVECT,@PPVEC          ;RESTORE TRAP RP04 VECTOR
 8472  033050  000000                          HALT                            ;CHANGE TO CONTINUE
 8473
 8474
 8475
 8476
```

```
                                                                                ;******************************
                                ;*********************************************************************
                                .SBTTL  SCOPE HANDLER ROUTINE

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

033052                          $SCOPE:
033052  006137  177570                  ROL     @#SWR           ;;LOOP ON PRESENT TEST?
033056  100511                         BMI     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
033060  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
033062  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
033066  012737  033106  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
033074  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
033100  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
033104  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
033106  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
033110  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
033114  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
033116                          6$:;#####END OF CODE FOR THE XOR TESTER#####
033116  032737  000400  177570          BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
033124  001404                         BEQ     2$              ;;BR IF NO
033126  123767  177570  145746          CMPB    @#SWR,$TSTNM    ;;ON THE RIGHT TEST?    SWR<7:0>
033134  001462                         BEQ     $OVER           ;;BR IF YES
033136  105767  145741          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
033142  001421                         BEQ     3$              ;;BR IF NO
033144  126767  145745  145731          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
033152  101015                         BHI     3$              ;;BR IF NO
033154  032737  001000  177570          BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
033162  001404                         BEQ     4$              ;;BR IF NO
033164  016767  145720  145714  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
033172  000443                         BR      $OVER
033174  105067  145703          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
033200  005067  146000                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
033204  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
033206  032737  004000  177570  3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
033214  001011                         BNE     1$              ;;BR IF YES
033216  005767  145656                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
033222  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
033224  005267  145654                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
033230  026767  145750  145646          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
033236  002021                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
033240  012767  000001  145636  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
033246  016767  000044  145730          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
033254  105267  145622          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
```

```
 8533  033260  011667  145622                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 8534  033264  011667  145620                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 8535  033270  005067  145712                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 8536  033274  112767  000001  145613        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 8537  033302  016737  145574  177570 $OVER: MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER
 8538  033310  016716  145572              MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 8539  033314  000002                      RTI                     ;;FIXES PS
 8540  033316  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
033320                          $TYPDS:
033320  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
033322  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033324  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033326  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
033330  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
033332  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
033336  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
033342  100004                         BPL     1$              ;;BR IF INPUT IS POS.
033344  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
033346  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
033354  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
033356  012703  033534                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
033362  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
033366  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
033370  016001  033524                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
033374  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
033376  002402                         BLT     4$              ;;BR IF DONE
033400  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
033402  000774                         BR      3$
033404  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
033406  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
033410  001002                         BNE     5$              ;;FALL THROUGH IF 0
033412  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
033414  100407                         BMI     7$              ;;BR IF YES
033416  106316                 5$:     ASLB    (SP)            ;;MSD?
033420  103003                         BCC     6$              ;;BR IF NO
033422  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
033430  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
033434  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
033440  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
033442  005720                         TST     (R0)+           ;;JUST INCREMENTING
033444  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
033450  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
033452  003002                         BGT     8$              ;;GO TO EXIT
033454  010502                         MOV     R5,R2           ;;GET THE LSD
033456  000764                         BR      6$              ;;GO CHANGE TO ASCII
033460  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
033462  100003                         BPL     9$              ;;BR IF NO
033464  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
033472  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
033474  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
033476  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
8597  033500  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
8598  033502  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8599  033504  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
8600  033506  104400  033534                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
8601  033512  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
8602  033520  012616                          MOV     (SP)+,(SP)
8603  033522  000002                          RTI                     ;;RETURN TO USER
8604  033524  023420                $DTBL:    10000.
8605  033526  001750                          1000.
8606  033530  000144                          100.
8607  033532  000012                          10.
8608  033534  000004                $DBLK:    .BLKW   4
8609                                 ;*********************************************************************
8610
8611                                 .SBTTL  TYPE ROUTINE
8612
8613                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
8614                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
8615                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
8616                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
8617                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
8618                                 ;*
8619                                 ;*CALL:
8620                                 ;*1) USING A TRAP INSTRUCTION
8621                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
8622                                 ;*OR
8623                                 ;*      TYPE
8624                                 ;*      MESADR
8625                                 ;*
8626                                 ;*2) USING A JSR INSTRUCTION
8627                                 ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
8628                                 ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
8629                                 ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
8630
8631  033544  105767  145401        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
8632  033550  100002                          BPL     1$              ;;BR IF YES
8633  033552  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
8634  033554  000407                          BR      3$              ;;LEAVE
8635  033556  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
8636  033560  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
8637  033564  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
8638  033566  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
8639  033570  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
8640  033572  012600                          MOV     (SP)+,R0        ;;RESTORE R0
8641  033574  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
8642  033600  000002                          RTI                     ;;RETURN
8643  033602  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
8644  033606  001424                          BEQ     8$
8645  033610  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
8646  033614  001004                          BNE     5$
8647  033616  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
8648  033620  104400  001215                  TYPE,   $CRLF
8649  033624  000757                          BR      2$              ;;GET NEXT CHARACTER
8650  033626  004767  000052          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
8651  033632  126726  145312          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
8652  033636  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
8653  033640  016746  145302            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
8654                                                              ;;AND THE NULL CHAR.
8655  033644  105366  000001      7$:   DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
8656  033650  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
8657  033652  004767  000026            JSR     PC,$TYPEC       ;;GO TYPE A NULL
8658  033656  000772                    BR      7$              ;;LOOP
8659
8660                              ;HORIZONTAL TAB PROCESSOR
8661
8662  033660  112716  000040      8$:   MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
8663  033664  004767  000014      9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
8664  033670  132767  000007  000052    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
8665  033676  001372                    BNE     9$              ;;TAB STOP
8666  033700  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
8667  033702  000730                    BR      2$              ;;GET NEXT CHARACTER
8668  033704  105777  145232    $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
8669  033710  100375                    BPL     $TYPEC
8670  033712  116677  000002  145224    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
8671  033720  122766  000015  000002    CMPB    #15,2(SP)       ;;BRANCH IF
8672  033726  001003                    BNE     1$              ;;NOT <CR>
8673  033730  105067  000014            CLRB    $CHARCNT        ;;
8674  033734  000406                    BR      $TYPEX          ;;EXIT
8675  033736  122766  000012  000002  1$:   CMPB    #12,2(SP)       ;;BRANCH IF
8676  033744  002002                    BGE     $TYPEX          ;;<LF>
8677  033746  105227                    INCB    (PC)+           ;;INC SPACE
8678  033750  000000            $CHARCNT:.WORD  0               ;;COUNT
8679  033752  000207            $TYPEX: RTS     PC
8680  033754  000207                    RTS     PC
8681                              ;;      EQUATES
8682          000200                    CRLF=   200             ;;<CR><LF> EQUIV
8683          000011                    HT=     11              ;;<HT>
8684
8685                              ;*******************************************************************
8686
8687                              .SBTTL  TTY INPUT ROUTINE
8688
8689  033756  000000            $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
8690  033760  000000            $TKQIN: .WORD   0               ;;INPUT POINTER
8691  033762  000000            $TKQOUT: .WORD  0               ;;OUTPUT POINTER
8692  033764  000011            $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
8693          033775            $TKQEND=.
8694          033776                    .EVEN
8695
8696                              ;*TK INITIALIZE ROUTINE
8697                              ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
8698                              ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
8699                              ;
8700                              ;*CALL:
8701                              ;*      JSR     PC,$TKINT
8702                              ;*      RETURN
8703                              ;
8704  033776  005067  177754    $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
8705  034002  012767  033764  177750    MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
8706  034010  016767  177744  177744    MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
8707  034016  012737  034046  000060    MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
8708  034024  012737  000200  000062    MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
```

```
 8709  034032  005777  145102          TST     @$TKB           ;;CLEAR DONE FLAG
 8710  034036  012777  000100  145072  MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
 8711  034044  000207                  RTS     PC              ;;RETURN TO CALLER
 8712
 8713                                  ;*TK SERVICE ROUTINE
 8714                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
 8715                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
 8716                                  ;*IT IN THE QUEUE.
 8717                                  ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
 8718                                  ;*UPON RETURN EXIT IS MAKE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
 8719                                  ;
 8720  034046  117746  145066  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
 8721  034052  042716  177600          BIC     #^C177,(SP)     ;;STRIP THE JUNK
 8722  034056  021627  000003          CMP     (SP),#3         ;;IS IT A CONTROL C?
 8723  034062  001006                  BNE     1$              ;;BRANCH IF NO
 8724  034064  104400  034357          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
 8725  034070  004767  177702          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
 8726  034074  000167  175076          JMP     OPERSEL         ;;CONTROL C RESTART
 8727  034100  022767  000011  177650  1$:     CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
 8728  034106  001004                  BNE     2$              ;;BRANCH IF NO
 8729  034110  104400  001210          TYPE    ,$BELL          ;;RING THE TTY BELL
 8730  034114  005726                  TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
 8731  034116  000415                  BR      3$              ;;EXIT
 8732  034120  005267  177632  2$:     INC     $TKCNT          ;;COUNT THIS CHARACTER
 8733  034124  112677  177630          MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 8734  034130  005267  177624          INC     $TKQIN          ;;UPDATE THE POINTER
 8735  034134  026727  177620  033775  CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 8736  034142  001003                  BNE     3$              ;;BRANCH IF NO
 8737  034144  012767  033764  177606  MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 8738  034152  000002          3$:     RTI                     ;;RETURN
 8739                                  ;*************************************************************
 8740                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 8741                                  ;*CALL:
 8742                                  ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
 8743                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 8744                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 8745                                  ;
 8746
 8747  034154  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
 8748  034156  016666  000004  000002  MOV     4(SP),2(SP)     ;;THE PS
 8749  034164  005066  000004          CLR     4(SP)           ;;GET READY FOR A CHARACTER
 8750  034170  005037  177776          CLR     @#PS            ;;ALLOW INTERRUPTS
 8751  034174  005767  177556  1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
 8752  034200  001775                  BEQ     1$
 8753  034202  005367  177550          DEC     $TKCNT          ;;DECREMENT THE COUNTER
 8754  034206  117766  177550  000004  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
 8755  034214  005267  177542          INC     $TKQOUT         ;;UPDATE THE POINTER
 8756  034220  026727  177536  033775  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
 8757  034226  001003                  BNE     2$              ;;BRANCH IF NO
 8758  034230  012767  033764  177524  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
 8759  034236  000002          2$:     RTI                     ;;RETURN
 8760                                  ;*************************************************************
 8761                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 8762                                  ;*CALL:
 8763                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 8764                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
```

```
8765                                    ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
8766
8767  034240  010346                    $PDLIN: MOV     R3,-(SP)        ;;SAVE R3
8768  034242  012703  034346            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
8769  034246  022703  034357            2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
8770  034252  101405                            BLOS    4$              ;;BR IF YES
8771  034254  104412                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
8772  034256  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
8773  034260  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
8774  034264  001003                            BNE     3$              ;;SKIP IF NOT
8775  034266  104400  001214            4$:     TYPE    ,$QUES          ;;TYPE A '?'
8776  034272  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
8777  034274  111367  000044            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
8778  034300  104400  034344                    TYPE    ,9$
8779  034304  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
8780  034310  001356                            BNE     2$              ;;LOOP IF NOT RETURN
8781  034312  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
8782  034316  104400  001216                    TYPE    ,$LF            ;;TYPE A LINE FEED
8783  034322  012603                            MOV     (SP)+,R3        ;;RESTORE R3
8784  034324  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
8785  034326  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
8786  034334  012766  034346  000004            MOV     #$TTYIN,4(SP)
8787  034342  000002                            RTI                     ;;RETURN
8788  034344     000                 9$:        .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
8789  034345     000                            .BYTE   0               ;;TERMINATOR
8790  034346  000011            $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
8791  034357     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
8792                                                                    ;FROM THE TTY
8793
8794
8795
8796                                    ;*******************************************************************
8797
8798                                    .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
8799
8800                                    ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
8801                                    ;*CHANGE IT TO BINARY.
8802                                    ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
8803                                    ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
8804                                    ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
8805                                    ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
8806                                    ;*CALL:
8807                                    ;*      RDOCT                   ;;READ AN OCTAL NUMBER
8808                                    ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
8809                                    ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
8810
8811  034364  011646                    $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
8812  034366  016666  000004  000002            MOV     4(SP),2(SP)     ;;INPUT NUMBER
8813  034374  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8814  034376  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8815  034400  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8816  034402  104414                    1$:     RDLIN                   ;;READ AN ASCIZ LINE
8817  034404  012600                            MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
8818  034406  010067  000100                    MOV     R0,5$           ;;AND SAVE IT
8819  034412  005001                            CLR     R1              ;;CLEAR DATA WORD
8820  034414  005002                            CLR     R2
```

```
8821  034416  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
8822  034420  001420                        BEQ     3$              ;;IF ZERO GET OUT
8823  034422  122716  000060                CMPB    #'0,(SP)        ;;MAKE SURE THIS CHAPACTER
8824  034426  003026                        BGT     4$              ;;IS AN OCTAL DIGIT
8825  034430  122716  000067                CMPB    #'7,(SP)
8826  034434  002423                        BLT     4$
8827  034436  006301                        ASL     R1              ;;*2
8828  034440  006102                        ROL     R2
8829  034442  006301                        ASL     R1              ;;*4
8830  034444  006102                        ROL     R2
8831  034446  006301                        ASL     R1              ;;*8
8832  034450  006102                        ROL     R2
8833  034452  042716  177770                BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
8834  034456  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
8835  034460  000756                        BR      2$              ;;LOOP
8836  034462  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
8837  034464  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
8838  034470  010267  000026                MOV     R2,$HIOCT
8839  034474  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8840  034476  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8841  034500  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8842  034502  000002                        RTI                     ;;RETURN
8843  034504  005726                4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
8844  034506  105010                        CLRB    (R0)            ;;SET A TERMINATOR
8845  034510  104400                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
8846  034512  000000                5$:     .WORD   0
8847  034514  104400  001214                TYPE    ,$QUES          ;;"?" "CR" & "LF"
8848  034520  000730                        BR      1$              ;;TRY AGAIN
8849  034522  000000              $HIOCT:   .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
8850                                ;**********************************************************************
8851
8852                                .SBTTL  ERROR HANDLER ROUTINE
8853
8854                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
8855                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8856                                ;*AND GO TO $ERRTYP ON ERROR
8857                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8858                                ;*SW15=1         HALT ON ERROR
8859                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
8860                                ;*SW10=1         BELL ON ERROR
8861                                ;*SW09=1         LOOP ON ERROR
8862                                ;*CALL
8863                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
8864
8865  034524                        $ERROR:
8866
8867
8868  034524                        REGSAV:
8869  034524  012737  177777  004616          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
8870  034532                        REGSA1:
8871
8872
8873  034532  105267  144345        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
8874  034536  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
8875  034540  016737  144336  177570          MOV     $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
8876  034546  032737  002000  177570          BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
8877  034554  001402                          BEQ     1$              ;;NO - SKIP
8878  034556  104400  001210                  TYPE    ,$BELL          ;;RING BELL
8879  034562  005267  144324        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
8880  034566  011667  144324                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
8881  034572  162767  000002  144316          SUB     #2,$ERRPC
8882  034600  117767  144312  144306          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
8883  034606  032737  020000  177570          BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
8884  034614  001004                          BNE     2$              ;;SKIP TYPEOUTS
8885  034616  004737  034670                  JSR     PC,@#$ERRTYP    ;;GO TO USER ERROR ROUTINE
8886  034622  104400  001215                  TYPE    ,$CRLF
8887  034626  005737  177570        2$:     TST     @#SWR           ;;HALT ON ERROR
8888  034632  100001                          BPL     3$              ;;SKIP IF CONTINUE
8889  034634  000000                          HALT                    ;;HALT ON ERROR!
8890  034636  032737  001000  177570  3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
8891  034644  001402                          BEQ     4$              ;;BR IF NO
8892  034646  016716  144236                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
8893  034652  005767  144330        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
8894  034656  001402                          BEQ     5$              ;;BR IF NONE
8895  034660  016716  144322                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
8896  034664                        5$:
8897  034664  000002                          RTI                     ;;RETURN
8898                                ;;**********************************************************
8899                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
8900                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
8901                                ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
8902                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
8903                                ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
8904                                ;*WITH ONLY MINOR CHANGES
8905                                ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
```

```
                                        ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
                                        ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
                                        ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
                                        ;*AND NOT THE ERROR MESSAGE AND HEADER.
034666  000000                          PRITEM: 0                       ;PREVIOUS ITEM NO. LOCATION
034670  013746  177570                  $ERRTYP: MOV    @#SWR,-(SP)     ;GET SWITCH SETTING
034674  042716  177277                          BIC    #^C500,(SP)     ;KEEP ONLY SWITCH 8 AND 6
034700  022726  000100                          CMP    #SW06,(SP)+     ;IS 6 SET AND 8 RESET
034704  001001                              BNE    1$              ;IF NOT BRANCH
034706  000402                              BR     2$              ;BRANCH IF SW 6 IS SET AND 8 RESET
034710  000137  035630                  1$:     JMP    @#TYPERR        ;JUMP IF SW 8 IS SET
                                                                    ;OR IF SW 8 IS RESET AND SW 6 IS RESET
034714                                  2$:

034714  104400  034722                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
034720  000406                              BR     64$             ;;GET OVER THE ASCIZ
                                            ;;.ASCIZ      <15><12>/RHWC =  /
034736                                  64$:
034736  013746  002242                          MOV    @#WC,-(SP)      ;GET READY TO TYPE RHWC CONTENTS
034742  104402                              TYPOC

034744  104400  034752                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
034750  000406                              BR     65$             ;;GET OVER THE ASCIZ
                                            ;;.ASCIZ      <15><12>/RHBA =  /
034766                                  65$:
034766  013746  002244                          MOV    @#BA,-(SP)      ;GET READY TO TYPE RHBA CONTENTS
034772  104402                              TYPOC

034774  104400  035002                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
035000  000406                              BR     66$             ;;GET OVER THE ASCIZ
                                            ;;.ASCIZ      <15><12>/RHCS2 =  /
035016                                  66$:
035016  013746  002246                          MOV    @#CS2,-(SP)     ;GET READY TO TYPE RHCS2 CONTENTS
035022  104402                              TYPOC

035024  104400  035032                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
035030  000406                              BR     67$             ;;GET OVER THE ASCIZ
                                            ;;.ASCIZ      <15><12>/RHCS1 =  /
035046                                  67$:
035046  013746  002250                          MOV    @#CS1,-(SP)     ;GET READY TO TYPE RHCS1 CONTENTS
035052  104402                              TYPOC

035054  104400  035062                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
035060  000406                              BR     68$             ;;GET OVER THE ASCIZ
                                            ;;.ASCIZ      <15><12>/RHDS1 =  /
035076                                  68$:
035076  013746  002272                          MOV    @#DS1,-(SP)     ;GET READY TO TYPE RHDS1 CONTENTS
035102  104402                              TYPOC

035104  104400  035112                          TYPE   ,.+4            ;;TYPE ASCIZ STRING
035110  000406                              BR     69$             ;;GET OVER THE ASCIZ
```

```
8962                                                    ;;.ASCIZ      <15><12>/RHER1 =  /
8963  035126                                    698$:
8964  035126  013746  002252                            MOV     @#ER1,-(SP)     ;GET READY TO TYPE RHER1 CONTENTS
8965  035132  104402                                    TYPOC
8966
8967
8968  035134  104400  035142                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
8969  035140  000406                                    BR      70$             ;;GET OVER THE ASCIZ
8970                                                    ;;.ASCIZ      <15><12>/RHER2 =  /
8971  035156                                    708$:
8972  035156  013746  002256                            MOV     @#ER2,-(SP)     ;GET READY TO TYPE RHER2 CONTENTS
8973  035162  104402                                    TYPOC
8974
8975
8976  035164  104400  035172                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
8977  035170  000406                                    BR      71$             ;;GET OVER THE ASCIZ
8978                                                    ;;.ASCIZ      <15><12>/RHER3 =  /
8979  035206                                    718$:
8980  035206  013746  002264                            MOV     @#ER3,-(SP)     ;GET READY TO TYPE RHER3 CONTENTS
8981  035212  104402                                    TYPOC
8982
8983
8984  035214  104400  035222                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
8985  035220  000406                                    BR      72$             ;;GET OVER THE ASCIZ
8986                                                    ;;.ASCIZ      <15><12>/RHDST =  /
8987  035236                                    728$:
8988  035236  013746  002254                            MOV     @#DST,-(SP)     ;GET READY TO TYPE RHDST CONTENTS
8989  035242  104402                                    TYPOC
8990
8991
8992  035244  104400  035252                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
8993  035250  000406                                    BR      73$             ;;GET OVER THE ASCIZ
8994                                                    ;;.ASCIZ      <15><12>/RHCA =  /
8995  035266                                    738$:
8996  035266  013746  002262                            MOV     @#CA,-(SP)      ;GET READY TO TYPE RHCA CONTENTS
8997  035272  104402                                    TYPOC
8998
8999
9000  035274  104400  035302                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
9001  035300  000406                                    BR      74$             ;;GET OVER THE ASCIZ
9002                                                    ;;.ASCIZ      <15><12>/RHAS =  /
9003  035316                                    748$:
9004  035316  013746  002266                            MOV     @#AS,-(SP)      ;GET READY TO TYPE RHAS CONTENTS
9005  035322  104402                                    TYPOC
9006
9007
9008  035324  104400  035332                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
9009  035330  000406                                    BR      75$             ;;GET OVER THE ASCIZ
9010                                                    ;;.ASCIZ      <15><12>/RHOF =  /
9011  035346                                    758$:
9012  035346  013746  002260                            MOV     @#OF,-(SP)      ;GET READY TO TYPE RHOF CONTENTS
9013  035352  104402                                    TYPOC
9014
9015
9016  035354  104400  035362                            TYPE    ,.+4            ;;TYPE ASCIZ STRING
9017  035360  000406                                    BR      76$             ;;GET OVER THE ASCIZ
```

```
                                        ;;.ASCIZ    <15><12>/RHMR =  /
035376                          76$:
035376  013746  002270                  MOV     @#MR,-(SP)      ;GET READY TO TYPE RHMR CONTENTS
035402  104402                          TYPOC
                                
                                
035404  104400  035412                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035410  000406                          BR      77$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHLA =  /
035426                          77$:
035426  013746  002306                  MOV     @#LA,-(SP)      ;GET READY TO TYPE RHLA CONTENTS
035432  104402                          TYPOC
                                
                                
035434  104400  035442                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035440  000406                          BR      78$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHCC =  /
035456                          78$:
035456  013746  002304                  MOV     @#CC,-(SP)      ;GET READY TO TYPE RHCC CONTENTS
035462  104402                          TYPOC
                                
                                
035464  104400  035472                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035470  000406                          BR      79$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHEC1 =  /
035506                          79$:
035506  013746  002300                  MOV     @#EC1,-(SP)     ;GET READY TO TYPE RHEC1 CONTENTS
035512  104402                          TYPOC
                                
                                
035514  104400  035522                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035520  000406                          BR      80$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHEC2 =  /
035536                          80$:
035536  013746  002302                  MOV     @#EC2,-(SP)     ;GET READY TO TYPE RHEC2 CONTENTS
035542  104402                          TYPOC
                                
                                
035544  104400  035552                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035550  000406                          BR      81$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHDT =  /
035566                          81$:
035566  013746  002274                  MOV     @#DT,-(SP)      ;GET READY TO TYPE RHDT CONTENTS
035572  104402                          TYPOC
                                
                                
035574  104400  035602                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
035600  000406                          BR      82$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ    <15><12>/RHSN =  /
035616                          82$:
035616  013746  002276                  MOV     @#SN,-(SP)      ;GET READY TO TYPE RHSN CONTENTS
035622  104402                          TYPOC
                                
035624  005037  034666                  CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
035630                          TYPERR:
035630  104400  001215                  TYPE    ,SCRLF          ;"CARRIAGE RETURN" & "LINE FEED"
```

```
 9074  035634  010046                        MOV     R0,-(SP)        ;SAVE R0
 9075  035636  005000                        CLR     R0              ;PICKUP THE ITEM INDEX
 9076  035640  153700  001114                BISB    @#$ITEMB,R0
 9077  035644  001004                        BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
 9078                                                                ;TYPE THE PC OF THE ERROR
 9079  035646  016746  143244                MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
 9080                                                                ;ERROR ADDRESS
 9081  035652  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9082  035654  000454                        BR      10$             ;GET OUT
 9083  035656  005300                1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
 9084  035660  006300                        ASL     R0              ;       WORK FOR THE ERROR TABLE
 9085  035662  006300                        ASL     R0
 9086  035664  006300                        ASL     R0
 9087  035666  062700  001220                ADD     #$EPRTB,R0      ;FORM TABLE POINTER
 9088  035672  020037  034666                CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
 9089  035676  001002                        BNE     13$             ;BRANCH IF NOT
 9090  035700  022020                        CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
 9091  035702  000420                        BR      5$
 9092  035704  010037  034666        13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
 9093  035710  012067  000004                MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
 9094  035714  001404                        BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
 9095  035716  104400                        TYPE                    ;TYPE THE "ERROR MESSAGE"
 9096  035720  000000                2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
 9097  035722  104400  001215                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
 9098  035726  012067  000004        3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
 9099  035732  001404                        BEQ     5$              ;SKIP TYPEOUT IF 0
 9100  035734  104400                        TYPE                    ;TYPE THE "DATA HEADER"
 9101  035736  000000                4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
 9102  035740  104400  001215                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
 9103  035744  010146                5$:     MOV     R1,-(SP)        ;SAVE R1
 9104  035746  012001                        MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
 9105  035750  001415                        BEQ     9$              ;BR IF NO DATA TO BE TYPED
 9106  035752  012000                        MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
 9107  035754  105720                6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
 9108  035756  001003                        BNE     7$              ;BR IF DECIMAL
 9109  035760  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
 9110  035762  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 9111  035764  000402                        BR      8$
 9112  035766                        7$:
 9113  035766  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
 9114  035770  104410                        TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
 9115  035772  005711                8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
 9116  035774  001403                        BEQ     9$              ;BR IF NO
 9117  035776  104400  036012                TYPE    ,11$            ;TYPE TWO(2) SPACES
 9118  036002  000764                        BR      6$              ;LOOP
 9119
 9120  036004  012601                9$:     MOV     (SP)+,R1        ;RESTORE R1
 9121  036006  012600                10$:    MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
 9122  036010  000207                        RTS     PC              ;RETURN
 9123  036012  020040      000       11$:    .ASCIZ  /  /            ;TWO(2) SPACES
 9124          036016                        .EVEN
 9125                                ;*****************************************************************
 9126                                ;*********************************************************************************
 9127                                ;*****************************************************************
 9128
 9129                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
```

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

036016  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
036022  116667  000001  000211         MOVB    1(SP),$0FILL    ;;LOAD ZERO FILL SWITCH
036030  112667  000207                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
036034  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
036040  000406                         BR      $TYPON
036042  112767  000001  000171 $TYPOC: MOVB    #1,$0FILL       ;;SET THE ZERO FILL SWITCH
036050  112767  000006  000165         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
036056  112767  000005  000154 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
036064  010346                         MOV     R3,-(SP)        ;;SAVE R3
036066  010446                         MOV     R4,-(SP)        ;;SAVE R4
036070  010546                         MOV     R5,-(SP)        ;;SAVE R5
036072  116704  000145                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
036076  005404                         NEG     R4
036100  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
036104  110467  000132                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
036110  116704  000125                 MOVB    $0FILL,R4       ;;GET THE ZERO FILL SWITCH
036114  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
036120  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
036122  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
036124  000404                         BR      3$              ;;GO DO MSB
036126  006105                 2$:     ROL     R5              ;;FORM THIS DIGIT
036130  006105                         ROL     R5
036132  006105                         ROL     R5
036134  010503                         MOV     R5,R3
036136  006103                 3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
036140  105367  000076                 DECB    $OMODE          ;;TYPE THIS DIGIT?
036144  100016                         BPL     7$              ;;BR IF NO
036146  042703  177770                 BIC     #177770,R3      ;;GET RID OF JUNK
036152  001002                         BNE     4$              ;;TEST FOR 0
036154  005704                         TST     R4              ;;SUPPRESS THIS 0?
036156  001403                         BEQ     5$              ;;BR IF YES
036160  005204                 4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
036162  052703  000060                 BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
```

```
 9186  036166  052703  000040          5$:     BIS     #'0,R3          ;;MAKE ASCII IF NOT ALREADY
 9187  036172  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
 9188  036176  104400  036236                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 9189  036202  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
 9190  036206  003347                          BGT     2$              ;;BR IF MORE TO DO
 9191  036210  002402                          BLT     6$              ;;BR IF DONE
 9192  036212  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 9193  036214  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 9194  036216  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 9195  036220  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 9196  036222  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 9197  036224  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 9198  036232  012616                          MOV     (SP)+,(SP)
 9199  036234  000002                          RTI                     ;;RETURN
 9200  036236     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 9201  036237     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 9202  036240     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 9203  036241     000          $0FILL: .BYTE   0               ;;ZERO FILL SWITCH
 9204  036242  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                ;**********************************************************************
                                .SBTTL  TRAP DECODER

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

036244  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
036246  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
036252  005740                          TST     -(R0)           ;;BACKUP BY 2
036254  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
036256  016000  036264                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
036262  000200                          RTS     R0              ;;GO TO ROUTINE

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
036264                          $TRPAD:
036264  033544                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
036266  036042                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
036270  036016                          $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
036272  036056                          $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
036274  033320                          $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
036276  034154                          $RDCHR  ;;CALL=RDCHR    TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE
036300  034240                          $RDLIN  ;;CALL=RDLIN    TRAP+14(104414) TTY TYPEIN STRING ROUTINE
036302  034364                          $RDOCT  ;;CALL=RDOCT    TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY
036304  030552                          WAIT.T  ;;CALL=WAT      TRAP+20(104420) DONT ADD ABOVE THIS TRAP
```

```
 9242                                ;*****************************************************************
 9243
 9244                                .SBTTL  POWER DOWN AND UP ROUTINES
 9245
 9246                                ;POWER DOWN ROUTINE
 9247  036306  012737  036434  000024 $PWPDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 9248  036314  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9249  036322  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9250  036324  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9251  036326  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9252  036330  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9253  036332  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9254  036334  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9255  036336  010667  000076                MOV     SP,$SAVR6       ;;SAVE SP
 9256  036342  012737  036354  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 9257  036350  000000                        HALT
 9258  036352  000776                        BR      .-2             ;;HANG UP
 9259
 9260                                ;POWER UP ROUTINE
 9261  036354  016706  000060        $PWRUP: MOV     $SAVR6,SP       ;;GET SP
 9262  036360  005067  000054                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 9263  036364  005267  000050        1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 9264  036370  001375                        BNE     1$              ;;OF  WORD
 9265  036372  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9266  036374  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 9267  036376  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9268  036400  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9269  036402  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9270  036404  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9271  036406  012737  036306  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 9272  036414  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
 9273  036422  104400                        TYPE                    ;REPORT THE POWER FAILURE
 9274  036424  036442        $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
 9275  036426  012716                        MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
 9276  036430  004666        $PWRAD: .WORD   BEGIN                   ;;RESTART ADDRESS
 9277  036432  000002                        RTI
 9278  036434  000000        $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
 9279  036436  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 9280  036440  000000        $SAVR6: 0                               ;;PUT THE SP HERE
 9281  036442  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
 9282  036450  000122
 9283                                .EVEN
```

```
                                ;*****************************************************************************
                                ;
                                ;ERROR AND MESSAGE TABLE CONDIMENTS
                                ;
                                ;*****************************************************************************


036452  050122  032060  042040  EM1:    .ASCIZ  /RP04 DID NOT INTERRUPT/
036460  042111  047040  052117
036466  044440  052116  051105
036474  052522  052120     000
036501     111  052116  051105  EM2:    .ASCIZ  /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
036506  052522  052120  042440
036514  040516  046102  020105
036522  044502  020124  047504
036530  047127  041040  052125
036536  042440  050130  041505
036544  042524  020104  044502
036552  020124  044504  020104
036560  047516  020124  042523
036566  000124
036570  050122  032060  042040  EM3:    .ASCIZ  /RP04 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
036576  042111  047040  052117
036604  044440  052116  051105
036612  052522  052120  053440
036620  042510  020116  054105
036626  042520  052103  042105
036634  041040  052111  042040
036642  042111  051440  052105
036650     000
036651     105  050130  041505  EM4:    .ASCIZ  /EXPECTED BIT DID SET BUT TIME IS IN ERROR TIME IN 10 MICROSEC. DECIMAL/
036656  042524  020104  044502
036664  020124  044504  020104
036672  042523  020124  052502
036700  020124  044524  042515
036706  044440  020123  047111
036714  042440  051122  051117
036722  052040  046511  020105
036730  047111  030440  020060
036736  044515  051103  051517
036744  041505  020056  042504
036752  044503  040515  000114
036760  044122  051501  042040  EM5:    .ASCIZ  /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
036766  042517  020123  047516
036774  020124  046103  040505
037002  020122  054502  046440
037010  053117  047111  020107
037016  047111  040440  046114
037024  047440  042516  000123
037032  047514  042101  047111  EM6:    .ASCIZ  /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
037040  020107  044122  051105
037046  020061  047506  020122
037054  046101  020114  047125
037062  052111  020123  044504
```

```
 9340  037070  020104  047516  020124
 9341  037076  042523  020124  047101
 9342  037104  020131  044122  051501
 9343  037112  041040  052111  000123
 9344  037120  050123  041505  043111  EM7:    .ASCIZ  /SPECIFIED REGISTER NON EXISTANT, SO ABORT PROGRAM/
 9345  037126  042511  020104  042522
 9346  037134  044507  052123  051105
 9347  037142  047040  047117  042440
 9348  037150  044530  052123  047101
 9349  037156  026124  051440  020117
 9350  037164  041101  051117  020124
 9351  037172  051120  043517  040522
 9352  037200  000115
 9353  037202  052123  050117  042520  EM10:   .ASCIZ  /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
 9354  037210  020104  051104  053111
 9355  037216  020105  040510  020123
 9356  037224  047515  020114  044502
 9357  037232  020124  047111  051040
 9358  037240  042110  030523  051440
 9359  037246  052105     000
 9360  037251     127  052111  020110  EM11:   .ASCIZ  /WITH SPINDLE POWERED DOWN PHCS2 SHOULD ONLY HAVE UNIT NO: AND IR SET/
 9361  037256  050123  047111  046104
 9362  037264  020105  047520  042527
 9363  037272  042522  020104  047504
 9364  037300  047127  051040  041510
 9365  037306  031123  051440  047510
 9366  037314  046125  020104  047117
 9367  037322  054514  044040  053101
 9368  037330  020105  047125  052111
 9369  037336  047040  035117  040440
 9370  037344  042116  044440  020122
 9371  037352  042523  000124
 9372  037356  043101  042524  020122  EM12:   .ASCIZ  /AFTER SPINDLE POWERED UP, NO PACK ACKN, RHDS1 SHOULD HAVE MOL=1, VV=0/
 9373  037364  050123  047111  046104
 9374  037372  020105  047520  042527
 9375  037400  042522  020104  050125
 9376  037406  020054  047516  050040
 9377  037414  041501  020113  041501
 9378  037422  047113  020056  044122
 9379  037430  051504  020061  044123
 9380  037436  052517  042114  044040
 9381  037444  053101  020105  047515
 9382  037452  036514  026061  053040
 9383  037460  036526  000060
 9384  037464  044527  044124  051440  EM13:   .ASCIZ  /WITH SPINDLE POWERED, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, RDY=
 9385  037472  044520  042116  042514
 9386  037500  050040  053517  051105
 9387  037506  042105  020054  047516
 9388  037514  044440  052116  040511
 9389  037522  044514  042532  020054
 9390  037530  044122  051503  020061
 9391  037536  044123  052517  042114
 9392  037544  044040  053101  020105
 9393  037552  047507  030075  020054
 9394  037560  053104  036501  026061
 9395  037566  051040  054504  030475
```

```
 9396  037574  020054  042511  030075
 9397  037602     000
 9398  037603     101  052106  051105  EM14:   .ASCIZ  /AFTER SPINDLE POWERED UP RHCC SHOULD BE=0/
 9399  037610  051440  044520  042116
 9400  037616  042514  050040  053517
 9401  037624  051105  042105  052440
 9402  037632  020120  044122  041503
 9403  037640  051440  047510  046125
 9404  037646  020104  042502  030075
 9405  037654     000
 9406  037655     120  041501  020113  EM15:   .ASCII  /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
 9407  037662  041501  047113  053517
 9408  037670  042514  043504  020105
 9409  037676  047503  046515  047101
 9410  037704  020104  040503  051525
 9411  037712  042105  040440  020116
 9412  037720  051105  047522  006522
 9413  037726     012
 9414  037727     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
 9415  037734  040504  040524  044440
 9416  037742  020123  042502  047506
 9417  037750  042522  041440  046517
 9418  037756  040515  042116  020054
 9419  037764  042522  020103  040504
 9420  037772  040524  044440  020123
 9421  040000  043101  042524  020122
 9422  040006  047503  046515  047101
 9423  040014  000104
 9424  040016  047516  047455  020120  EM16:   .ASCII  /NO-OP COMMAND CAUSED AN ERROR/<15><12>
 9425  040024  047503  046515  047101
 9426  040032  020104  040503  051525
 9427  040040  042105  040440  020116
 9428  040046  051105  047522  006522
 9429  040054     012
 9430  040055     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
 9431  040062  040504  040524  044440
 9432  040070  020123  042502  047506
 9433  040076  042522  041440  046517
 9434  040104  040515  042116  020054
 9435  040112  042522  020103  040504
 9436  040120  040524  044440  020123
 9437  040126  043101  042524  020122
 9438  040134  047503  046515  047101
 9439  040142  000104
 9440  040144  051104  053111  020105  EM17:   .ASCII  /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
 9441  040152  046103  040505  020122
 9442  040160  047503  046515  047101
 9443  040166  020104  040503  051525
 9444  040174  042105  040440  020116
 9445  040202  051105  047522  006522
 9446  040210     012
 9447  040211     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
 9448  040216  040504  040524  043440
 9449  040224  053111  051505  051440
 9450  040232  047510  046125  020104
 9451  040240  042502  020054  042522
```

```
 9452  040246  020103  040504  040524
 9453  040254  043440  053111  051505
 9454  040262  040440  052106  051105
 9455  040270  041440  046517  040515
 9456  040276  042116     000
 9457  040301     122  040505  026504  EM20:   .ASCII  /READ-IN COMMAND CAUSED AN ERROR/<15><12>
 9458  040306  047111  041440  046517
 9459  040314  040515  042116  041440
 9460  040322  052501  042523  020104
 9461  040330  047101  042440  051122
 9462  040336  051117  005015
 9463  040342  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONTENTS AFTER COMMAND/
 9464  040350  052101  020101  044507
 9465  040356  042526  020123  044123
 9466  040364  052517  042114  041040
 9467  040372  026105  051040  041505
 9468  040400  042040  052101  020101
 9469  040406  044507  042526  020123
 9470  040414  042522  027107  041440
 9471  040422  047117  042524  052116
 9472  040430  020123  043101  042524
 9473  040436  020122  047503  046515
 9474  040444  047101  000104
 9475  040450  044122  051503  020061  EM21:   .ASCIZ  /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
 9476  040456  047503  052116  047105
 9477  040464  051524  042040  051125
 9478  040472  047111  020107  047503
 9479  040500  046515  047101  020104
 9480  040506  040527  020123  047111
 9481  040514  042440  051122  051117
 9482  040522     000
 9483  040523     122  042110  030523  EM22:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
 9484  040530  041440  047117  042524
 9485  040536  052116  020123  052504
 9486  040544  044522  043516  041440
 9487  040552  046517  040515  042116
 9488  040560  053440  051501  044440
 9489  040566  020116  051105  047522
 9490  040574  000122
 9491  040576  047125  047514  042101  EM23:   .ASCII  /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
 9492  040604  041440  046517  040515
 9493  040612  042116  041440  052501
 9494  040620  042523  020104  047101
 9495  040626  042440  051122  051117
 9496  040634  005015
 9497  040636  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REGISTER CONT. AFTER COMMAND/
 9498  040644  052101  020101  044507
 9499  040652  042526  020123  044123
 9500  040660  052517  042114  041040
 9501  040666  026105  051040  041505
 9502  040674  042040  052101  020101
 9503  040702  044507  042526  020123
 9504  040710  042522  044507  052123
 9505  040716  051105  041440  047117
 9506  040724  027124  040440  052106
 9507  040732  051105  041440  046517
```

```
9508  040740  040515  042116     000
9509  040745     117  043106  042523  EM24:   .ASCII  /OFFSET COMMAND CAUSED AN ERROR/<15><12>
9510  040752  020124  047503  046515
9511  040760  047101  020104  040503
9512  040766  051525  042105  040440
9513  040774  020116  051105  047522
9514  041002  006522     012
9515  041005     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
9516  041012  040504  040524  043440
9517  041020  053111  051505  051440
9518  041026  047510  046125  020104
9519  041034  042502  020054  042522
9520  041042  020103  040504  040524
9521  041050  043440  053111  051505
9522  041056  051040  043505  020056
9523  041064  047503  052116  020056
9524  041072  043101  042524  020122
9525  041100  047503  046515  047101
9526  041106  000104
9527  041110  042522  052524  047122  EM25:   .ASCII  /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
9528  041116  052040  020117  042503
9529  041124  052116  051105  046040
9530  041132  047111  020105  047503
9531  041140  046515  047101  020104
9532  041146  040503  051525  042105
9533  041154  040440  020116  051105
9534  041162  047522  006522     012
9535  041167     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
9536  041174  040504  040524  043440
9537  041202  053111  051505  051440
9538  041210  047510  046125  020104
9539  041216  042502  020054  042522
9540  041224  020103  040504  040524
9541  041232  043440  053111  051505
9542  041240  051040  043505  020056
9543  041246  047503  052116  020056
9544  041254  043101  042524  020122
9545  041262  047503  046515  047101
9546  041270  000104
9547  041272  030065  020060  043117  EM26:   .ASCIZ  /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
9548  041300  051506  052105  041440
9549  041306  046517  040515  042116
9550  041314  020123  047117  020105
9551  041322  043101  042524  020122
9552  041330  044124  020105  052117
9553  041336  042510  020122  040503
9554  041344  051525  042105  040440
9555  041352  020116  051105  047522
9556  041360  000122
9557  041362  051127  052111  020105  EM27:   .ASCII  /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
9558  041370  042510  042101  051105
9559  041376  040440  042116  042040
9560  041404  052101  020101  040503
9561  041412  051525  042105  040440
9562  041420  050115  047522  042520
9563  041426  020122  042522  044507
```

```
 9564  041434  052123  051105  041440
 9565  041442  040510  043516  006505
 9566  041450     012
 9567  041451     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9568  041456  040504  040524  043440
 9569  041464  053111  051505  053440
 9570  041472  040510  020124  044123
 9571  041500  052517  042114  041040
 9572  041506  020105  044124  051105
 9573  041514  006505     012
 9574  041517     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
 9575  041524  042526  020104  040504
 9576  041532  040524  043440  053111
 9577  041540  051505  053440  040510
 9578  041546  020124  040527  020123
 9579  041554  044124  051105  020105
 9580  041562  043101  042524  020122
 9581  041570  047503  046515  047101
 9582  041576  000104
 9583  041600  051127  052111  020105  EM30:   .ASCIZ  /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
 9584  041606  042510  042101  051105
 9585  041614  040440  042116  042040
 9586  041622  052101  020101  044103
 9587  041630  047101  042507  020104
 9588  041636  051127  052111  020105
 9589  041644  051106  046517  041040
 9590  041652  043125  042506  000122
 9591  041660  042522  042101  044040  EM31:   .ASCII  /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9592  041666  040505  042504  020122
 9593  041674  047101  020104  040504
 9594  041702  040524  041440  052501
 9595  041710  042523  020104  046511
 9596  041716  051120  050117  051105
 9597  041724  051040  043505  051511
 9598  041732  042524  020122  044103
 9599  041740  047101  042507  005015
 9600  041746  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9601  041754  052101  020101  044507
 9602  041762  042526  020123  044127
 9603  041770  052101  051440  047510
 9604  041776  046125  020104  042502
 9605  042004  052040  042510  042522
 9606  042012  005015
 9607  042014  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
 9608  042022  042105  042040  052101
 9609  042030  020101  044507  042526
 9610  042036  020123  044127  052101
 9611  042044  053440  051501  052040
 9612  042052  042510  042522  040440
 9613  042060  052106  051105  041440
 9614  042066  046517  040515  042116
 9615  042074     000
 9616  042075     127  044522  042524  EM32:   .ASCIZ  /WRITE HEADER DATA FOLLOWED BY READ HEADER AND DATA CAUSED DATA ERROR/
 9617  042102  044040  040505  042504
 9618  042110  020122  040504  040524
 9619  042116  043040  046117  047514
```

```
 9620  042124  042527  020104  054502
 9621  042132  051040  040505  020104
 9622  042140  042510  042101  051105
 9623  042146  040440  042116  042040
 9624  042154  052101  020101  040503
 9625  042162  051525  042105  042040
 9626  042170  052101  020101  051105
 9627  042176  047522  000122
 9628  042202  042522  042101  042040  EM33:   .ASCII  /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9629  042210  052101  020101  040503
 9630  042216  051525  042105  044440
 9631  042224  050115  047522  042520
 9632  042232  020122  042522  044507
 9633  042240  052123  051105  041440
 9634  042246  040510  043516  006505
 9635  042254     012
 9636  042255     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9637  042262  040504  040524  043440
 9638  042270  053111  051505  053440
 9639  042276  040510  020124  044123
 9640  042304  052517  042114  041040
 9641  042312  020105  044124  051105
 9642  042320  006505     012
 9643  042323     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
 9644  042330  042526  020104  040504
 9645  042336  040524  043440  053111
 9646  042344  051505  053440  040510
 9647  042352  020124  040527  020123
 9648  042360  044124  051105  020105
 9649  042366  043101  042524  020122
 9650  042374  047503  046515  047101
 9651  042402  000104
 9652  042404  042522  042101  042040  EM34:   .ASCIZ  /READ DATA INCORRECT/
 9653  042412  052101  020101  047111
 9654  042420  047503  051122  041505
 9655  042426  000124
 9656  042430  051127  052111  020105  EM35:   .ASCII  /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9657  042436  040504  040524  041440
 9658  042444  046517  040515  042116
 9659  042452  041440  052501  042523
 9660  042460  020104  046511  051120
 9661  042466  050117  051105  051040
 9662  042474  043505  051511  042524
 9663  042502  020122  044103  047101
 9664  042510  042507  005015
 9665  042514  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9666  042522  052101  020101  044507
 9667  042530  042526  020123  044127
 9668  042536  052101  051440  047510
 9669  042544  046125  020104  042502
 9670  042552  052040  042510  042522
 9671  042560  005015
 9672  042562  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
 9673  042570  042105  042040  052101
 9674  042576  020101  044507  042526
 9675  042604  020123  042522  044507
```

```
 9676  042612  052123  051105  041440
 9677  042620  047117  042524  052116
 9678  042626  020123  043101  042524
 9679  042634  020122  047503  046515
 9680  042642  047101  000104
 9681  042646  051127  052111  020105  EM36:   .ASCIZ  /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
 9682  042654  040504  040524  041440
 9683  042662  046517  040515  042116
 9684  042670  041440  040510  043516
 9685  042676  042105  053440  044522
 9686  042704  042524  043040  047522
 9687  042712  020115  052502  043106
 9688  042720  051105     000
 9689  042723     123  042505  020113  EM37:   .ASCII  /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9690  042730  047503  046515  047101
 9691  042736  020104  040503  051525
 9692  042744  042105  044440  050115
 9693  042752  047522  042520  020122
 9694  042760  042522  044507  052123
 9695  042766  051105  041440  040510
 9696  042774  043516  006505     012
 9697  043001     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9698  043006  040504  040524  043440
 9699  043014  053111  051505  053440
 9700  043022  040510  020124  044123
 9701  043030  052517  042114  041040
 9702  043036  020105  044124  051105
 9703  043044  006505     012
 9704  043047     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
 9705  043054  042526  020104  040504
 9706  043062  040524  043440  053111
 9707  043070  051505  051040  043505
 9708  043076  051511  042524  020122
 9709  043104  047503  052116  047105
 9710  043112  051524  040440  052106
 9711  043120  051105  051440  042505
 9712  043126  020113  047503  046515
 9713  043134  047101  000104
 9714  043140  051127  052111  020105  EM40:   .ASCII  /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9715  043146  044103  041505  020113
 9716  043154  040503  051525  042105
 9717  043162  044440  050115  047522
 9718  043170  042520  020122  042522
 9719  043176  044507  052123  051105
 9720  043204  041440  040510  043516
 9721  043212  006505     012
 9722  043215     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9723  043222  040504  040524  043440
 9724  043230  053111  051505  053440
 9725  043236  040510  020124  044123
 9726  043244  052517  042114  041040
 9727  043252  020105  044124  051105
 9728  043260  006505     012
 9729  043263     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
 9730  043270  042526  020104  040504
 9731  043276  040524  043440  053111
```

```
 9732  043304  051505  051040  043505
 9733  043312  051511  042524  020122
 9734  043320  047503  052116  047105
 9735  043326  051524  040440  052106
 9736  043334  051105  041440  046517
 9737  043342  040515  042116     000
 9738  043347     114  041517  044513  EM41:   .ASCII  /LOCKING OUT WPITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
 9739  043354  043516  047440  052125
 9740  043362  053440  044522  042524
 9741  043370  041040  020131  051127
 9742  043376  052111  020105  047514
 9743  043404  045503  041040  052125
 9744  043412  047524  020116  040503
 9745  043420  051525  042105  044440
 9746  043426  050115  047522  042520
 9747  043434  020122  042522  044507
 9748  043442  052123  051105  041440
 9749  043450  040510  043516  006505
 9750  043456     012
 9751  043457     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9752  043464  040504  040524  043440
 9753  043472  053111  051505  053440
 9754  043500  040510  020124  044123
 9755  043506  052517  042114  041040
 9756  043514  020105  044124  051105
 9757  043522  006505     012
 9758  043525     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
 9759  043532  042526  020104  040504
 9760  043540  040524  043440  053111
 9761  043546  051505  051040  043505
 9762  043554  051511  042524  020122
 9763  043562  047503  052116  047105
 9764  043570  051524  040440  052106
 9765  043576  051105  053440  044522
 9766  043604  042524  020123  042527
 9767  043612  042522  046040  041517
 9768  043620  042513  020104  052517
 9769  043626  000124
 9770  043630  052101  042524  050115  EM42:   .ASCII  /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
 9771  043636  044524  043516  052040
 9772  043644  020117  051127  052111
 9773  043652  020105  044527  044124
 9774  043660  053440  044522  042524
 9775  043666  020123  047514  045503
 9776  043674  042105  047440  052125
 9777  043702  041440  052501  042523
 9778  043710  020104  046511  051120
 9779  043716  050117  051105  051040
 9780  043724  043505  051511  042524
 9781  043732  020122  044103  047101
 9782  043740  042507  005015
 9783  043744  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9784  043752  052101  020101  044507
 9785  043760  042526  020123  044127
 9786  043766  052101  051440  047510
 9787  043774  046125  020104  042502
```

```
 9788  044002  052040  042510  042522
 9789  044010  005015
 9790  044012  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
 9791  044020  042105  042040  052101
 9792  044026  020101  044507  042526
 9793  044034  020123  042522  044507
 9794  044042  052123  051105  041440
 9795  044050  047117  042524  052116
 9796  044056  020123  043101  042524
 9797  044064  020122  052101  042524
 9798  044072  050115  042524  020104
 9799  044100  051127  052111  000105
 9800  044106  051127  052111  047111  EM43:   .ASCII  /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>
 9801  044114  020107  044527  044124
 9802  044122  053440  044522  042524
 9803  044130  020123  047514  045503
 9804  044136  042105  047440  052125
 9805  044144  041440  040510  043516
 9806  044152  042105  042040  051511
 9807  044160  020113  040504  040524
 9808  044166  005015
 9809  044170  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE WITH WRITE LOCKED OUT/<15
 9810  044176  052101  020101  044507
 9811  044204  042526  020123  044127
 9812  044212  052101  053440  051501
 9813  044220  047440  020116  044504
 9814  044226  045523  041040  043105
 9815  044234  051117  020105  051127
 9816  044242  052111  020105  044527
 9817  044250  044124  053440  044522
 9818  044256  042524  046040  041517
 9819  044264  042513  020104  052517
 9820  044272  006524     012
 9821  044275     127  051501  040440          .ASCII  /WAS ATTEMPTED/<15><12>
 9822  044302  052124  046505  052120
 9823  044310  042105  005015
 9824  044314  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
 9825  044322  042105  042040  052101
 9826  044330  020101  044507  042526
 9827  044336  020123  044127  052101
 9828  044344  053440  051501  051040
 9829  044352  040505  020104  040502
 9830  044360  045503  040440  052106
 9831  044366  051105  053440  044522
 9832  044374  042524  005015
 9833  044400  044527  044124  053440          .ASCIZ  /WITH WRITE LOCKED OUT WAS ATTEMPTED/
 9834  044406  044522  042524  046040
 9835  044414  041517  042513  020104
 9836  044422  052517  020124  040527
 9837  044430  020123  052101  042524
 9838  044436  050115  042524  000104
 9839  044444  047105  041101  044514  EM44:   .ASCII  /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
 9840  044452  043516  053440  044522
 9841  044460  042524  020123  054502
 9842  044466  053440  044522  042524
 9843  044474  046040  041517  020113
```

```
9844  044502  052502  052124  047117
9845  044510  041440  052501  042523
9846  044516  020104  046511  051120
9847  044524  050117  051105  051040
9848  044532  043505  051511  042524
9849  044540  020122  044103  047101
9850  044546  042507  005015
9851  044552  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9852  044560  052101  020101  044507
9853  044566  042526  020123  044127
9854  044574  052101  051440  047510
9855  044602  046125  020104  042502
9856  044610  052040  042510  042522
9857  044616  005015
9858  044620  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
9859  044626  042105  042040  052101
9860  044634  020101  044507  042526
9861  044642  020123  042522  044507
9862  044650  052123  051105  041440
9863  044656  047117  042524  052116
9864  044664  020123  043101  042524
9865  044672  020122  051127  052111
9866  044700  020105  047514  045503
9867  044706  041040  052125  047524
9868  044714  006516     012
9869  044717     105  040516  046102          .ASCIZ  /ENABLED WRITES/
9870  044724  042105  053440  044522
9871  044732  042524  000123
9872  044736  051124  047101  043123  EM45:   .ASCII  /TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 10/<15><12>
9873  044744  051105  044522  043516
9874  044752  047440  020116  040514
9875  044760  052123  041040  047514
9876  044766  045503  026440  041440
9877  044774  046131  047111  042504
9878  045002  020122  030464  026060
9879  045010  051440  041505  047524
9880  045016  020122  030462  020054
9881  045024  051124  041501  020113
9882  045032  034061  005015
9883  045036  040503  051525  042105          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
9884  045044  044440  050115  047522
9885  045052  042520  020122  042522
9886  045060  044507  052123  051105
9887  045066  041440  040510  043516
9888  045074  006505     012
9889  045077     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
9890  045104  040504  040524  043440
9891  045112  053111  051505  053440
9892  045120  040510  020124  044123
9893  045126  052517  042114  041040
9894  045134  020105  044124  051105
9895  045142  006505     012
9896  045145     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
9897  045152  042526  020104  040504
9898  045160  040524  043440  053111
9899  045166  051505  051040  043505
```

```
 9900  045174  051511  042524  020122
 9901  045202  047503  052116  047105
 9902  045210  051524  040440  052106
 9903  045216  051105  052040  040522
 9904  045224  051516  042506  000122
 9905  045232  040504  040524  051040  EM46:   .ASCII  /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 10/<15><12>
 9906  045240  040505  020104  051106
 9907  045246  046517  046040  051501
 9908  045254  020124  046102  041517
 9909  045262  020113  020055  054503
 9910  045270  044514  042116  051105
 9911  045276  032040  030061  020054
 9912  045304  042523  052103  051117
 9913  045312  031040  026061  052040
 9914  045320  040522  045503  030440
 9915  045326  006470     012
 9916  045331     111  020123  047111          .ASCIZ  /IS IN ERROR/
 9917  045336  042440  051122  051117
 9918  045344     000
 9919  045345     124  040522  051516  EM47:   .ASCII  /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
 9920  045352  042506  051122  047111
 9921  045360  020107  040504  040524
 9922  045366  043040  047522  020115
 9923  045374  047516  042516  044530
 9924  045402  052123  047101  020124
 9925  045410  042523  052103  051117
 9926  045416  041440  052501  042523
 9927  045424  020104  046511  051120
 9928  045432  050117  051105  006440
 9929  045440     012
 9930  045441     122  043505  051511          .ASCII  /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9931  045446  042524  020122  044103
 9932  045454  047101  042507  020054
 9933  045462  047507  042117  042040
 9934  045470  052101  020101  044507
 9935  045476  042526  020123  044127
 9936  045504  052101  051440  047510
 9937  045512  046125  020104  042502
 9938  045520  052040  042510  042522
 9939  045526  005015
 9940  045530  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
 9941  045536  042105  042040  052101
 9942  045544  020101  044507  042526
 9943  045552  020123  042522  044507
 9944  045560  052123  051105  041440
 9945  045566  047117  042524  052116
 9946  045574  020123  043101  042524
 9947  045602  020122  052101  042524
 9948  045610  050115  042524  020104
 9949  045616  051124  047101  043123
 9950  045624  051105     000
 9951  045627     124  040522  051516  EM50:   .ASCII  /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
 9952  045634  042506  051122  047111
 9953  045642  020107  051106  046517
 9954  045650  047040  047117  054105
 9955  045656  051511  040524  052116
```

```
 9956  045664  051440  041505  047524
 9957  045672  020122  040503  051525
 9958  045700  042105  042040  052101
 9959  045706  020101  051105  047522
 9960  045714  006522     012
 9961  045717     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9962  045724  040504  040524  043440
 9963  045732  053111  051505  053440
 9964  045740  040510  020124  044123
 9965  045746  052517  042114  041040
 9966  045754  020105  044124  051105
 9967  045762  006505     012
 9968  045765     102  042101  042040          .ASCIZ  /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
 9969  045772  052101  020101  044507
 9970  046000  042526  020123  044127
 9971  046006  052101  053440  051501
 9972  046014  044440  020116  052502
 9973  046022  043106  051105  040440
 9974  046030  052106  051105  052040
 9975  046036  040522  051516  042506
 9976  046044  000122
 9977  046046  044507  044526  043516  EM51:   .ASCII  /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
 9978  046054  044440  046114  043505
 9979  046062  046101  043040  047125
 9980  046070  052103  047511  020116
 9981  046076  040503  051525  042105
 9982  046104  044440  050115  047522
 9983  046112  042520  020122  042522
 9984  046120  044507  052123  051105
 9985  046126  041440  040510  043516
 9986  046134  006505     012
 9987  046137     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 9988  046144  040504  040524  043440
 9989  046152  053111  051505  053440
 9990  046160  040510  020124  044123
 9991  046166  052517  042114  041040
 9992  046174  020105  044124  051105
 9993  046202  006505     012
 9994  046205     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
 9995  046212  042526  020104  040504
 9996  046220  040524  043440  053111
 9997  046226  051505  051040  043505
 9998  046234  051511  042524  020122
 9999  046242  047503  052116  047105
10000  046250  051524  040440  052106
10001  046256  051105  044440  046114
10002  046264  043505  046101  043040
10003  046272  047125  052103  047511
10004  046300  020116  051511  043440
10005  046306  053111  047105     000
10006  046313     127  044522  042524  EM52:   .ASCII  /WRITE DATA ON NONEXISTANT SECTOR CAUSED IMPROPER REGISTER CHANGE/<15><1
10007  046320  042040  052101  020101
10008  046326  047117  047040  047117
10009  046334  054105  051511  040524
10010  046342  052116  051440  041505
10011  046350  047524  020122  040503
```

```
10012  046356  051525  042105  044440
10013  046364  050115  047522  042520
10014  046372  020122  042522  044507
10015  046400  052123  051105  041440
10016  046406  040510  043516  006505
10017  046414     012
10018  046415     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10019  046422  040504  040524  043440
10020  046430  053111  051505  053440
10021  046436  040510  020124  044123
10022  046444  052517  042114  041040
10023  046452  020105  044124  051105
10024  046460  006505     012
10025  046463     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
10026  046470  042526  020104  040504
10027  046476  040524  043440  053111
10028  046504  051505  051040  043505
10029  046512  051511  042524  020122
10030  046520  047503  052116  047105
10031  046526  051524  040440  052106
10032  046534  051105  040440  052124
10033  046542  046505  052120  042105
10034  046550  053440  044522  042524
10035  046556  042040  052101  000101
10036  046564  042522  042101  044040  EM53:   .ASCIZ  /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
10037  046572  040505  042504  020122
10038  046600  047101  020104  040504
10039  046606  040524  040440  052106
10040  046614  051105  040440  051440
10041  046622  040505  041522  020110
10042  046630  040503  051525  042105
10043  046636  042040  052101  020101
10044  046644  051105  047522  000122
10045  046652  052101  042524  050115  EM54:   .ASCII  /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
10046  046660  044524  043516  041440
10047  046666  046517  040515  042116
10048  046674  053440  052111  020110
10049  046702  047111  040526  044514
10050  046710  020104  042101  051104
10051  046716  051505  020123  040503
10052  046724  051525  042105  044440
10053  046732  050115  047522  042520
10054  046740  020122  042522  044507
10055  046746  052123  051105  041440
10056  046754  040510  043516  006505
10057  046762     012
10058  046763     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10059  046770  040504  040524  043440
10060  046776  053111  051505  053440
10061  047004  040510  020124  044123
10062  047012  052517  042114  041040
10063  047020  020105  044124  051105
10064  047026  006505     012
10065  047031     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
10066  047036  042526  020104  040504
10067  047044  040524  043440  053111
```

```
10068  047052  051505  051040  043505
10069  047060  051511  042524  020122
10070  047066  047503  052116  047105
10071  047074  051524  040440  052106
10072  047102  051105  047440  042520
10073  047110  040522  044524  047117
10074  047116     000
10075  047117     127  044522  044524  EM55:   .ASCII  /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
10076  047124  043516  047440  020122
10077  047132  042522  042101  047111
10078  047140  020107  044527  044124
10079  047146  042440  050130  041505
10080  047154  042524  020104  042101
10081  047162  051104  051505  020123
10082  047170  053117  051105  046106
10083  047176  053517  042440  051122
10084  047204  051117  005015
10085  047210  040503  051525  042105          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
10086  047216  044440  050115  047522
10087  047224  042520  020122  042522
10088  047232  044507  052123  051105
10089  047240  041440  040510  043516
10090  047246  006505     012
10091  047251     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10092  047256  040504  040524  043440
10093  047264  053111  051505  053440
10094  047272  040510  020124  044123
10095  047300  052517  042114  041040
10096  047306  020105  044124  051105
10097  047314  006505     012
10098  047317     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
10099  047324  042526  020104  040504
10100  047332  040524  043440  0531.1
10101  047340  051505  051040  043505
10102  047346  051511  042524  020122
10103  047354  047503  052116  047105
10104  047362  051524  040440  052106
10105  047370  051105  047440  042520
10106  047376  040522  044524  047117
10107  047404     000
10108  047405     104  052101  020101  EM56:   .ASCII  /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
10109  047412  042522  042101  053440
10110  047420  052111  020110  047101
10111  047426  042440  050130  041505
10112  047434  042524  020104  042101
10113  047442  051104  051505  020123
10114  047450  053117  051105  046106
10115  047456  053517  042440  051122
10116  047464  051117  044440  020123
10117  047472  047111  047503  051122
10118  047500  041505  006524     012
10119  047505     127  051117  020104          .ASCII  /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>
10120  047512  047516  020056  020061
10121  047520  047524  031040  030066
10122  047526  051440  047510  046125
10123  047534  020104  042502  051040
```

```
10124  047542  040505  026104  053440
10125  047550  051117  020104  047516
10126  047556  031040  030466  052040
10127  047564  020117  033062  020066
10128  047572  044123  052517  042114
10129  047600  005015
10130  047602  042502  041440  040510          .ASCIZ  /BE CHANGED/
10131  047610  043516  042105     000
10132  047615     101  052124  046505  EM57:   .ASCII  /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
10133  047622  052120  047111  020107
10134  047630  040504  040524  041440
10135  047636  046517  040515  042116
10136  047644  053440  052111  020110
10137  047652  051127  047117  020107
10138  047660  047506  046522  052101
10139  047666  041040  052111  041440
10140  047674  052501  042523  006504
10141  047702     012
10142  047703     111  050115  047522          .ASCII  /IMPROPER REGISTER CHANGE/<15><12>
10143  047710  042520  020122  042522
10144  047716  044507  052123  051105
10145  047724  041440  040510  043516
10146  047732  006505     012
10147  047735     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10148  047742  040504  040524  043440
10149  047750  053111  051505  053440
10150  047756  040510  020124  044123
10151  047764  052517  042114  041040
10152  047772  020105  044124  051105
10153  050000  006505     012
10154  050003     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
10155  050010  042526  020104  040504
10156  050016  040524  043440  053111
10157  050024  051505  051040  043505
10158  050032  051511  042524  020122
10159  050040  047503  052116  047105
10160  050046  051524  040440  052106
10161  050054  051105  040440  052124
10162  050062  046505  052120  042105
10163  050070  042040  052101  020101
10164  050076  051124  047101  043123
10165  050104  051105     000
10166  050107     101  052124  046505  EM60:   .ASCII  /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
10167  050114  052120  047111  020107
10168  050122  047524  046440  042117
10169  050130  043111  020131  042522
10170  050136  044507  052123  051105
10171  050144  042040  051125  047111
10172  050152  020107  047101  047440
10173  050160  042520  040522  044524
10174  050166  047117  041440  052501
10175  050174  042523  020104  046511
10176  050202  051120  050117  051105
10177  050210  005015
10178  050212  042522  044507  052123          .ASCII  /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10179  050220  051105  041440  040510
```

```
10180  050226  043516  027105  043440
10181  050234  047517  020104  040504
10182  050242  040524  043440  053111
10183  050250  051505  053440  040510
10184  050256  020124  044123  052517
10185  050264  042114  041040  020105
10186  050272  044124  051105  006505
10187  050300     012
10188  050301     122  041505  044505          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
10189  050306  042526  020104  040504
10190  050314  040524  043440  053111
10191  050322  051505  051040  043505
10192  050330  051511  042524  020122
10193  050336  047503  052116  047105
10194  050344  051524  040440  052106
10195  050352  051105  047440  042520
10196  050360  040522  044524  047117
10197  050366  053440  051501  040440
10198  050374  052124  046505  052120
10199  050402  042105  005015
10200  050406  047515  043104  047111          .ASCIZ  /MODFING REG GIVES ADDRESS OF REGISTER BEING MODIFYED WHICH CAUSED ERROR
10201  050414  020107  042522  020107
10202  050422  044507  042526  020123
10203  050430  042101  051104  051505
10204  050436  020123  043117  051040
10205  050444  043505  051511  042524
10206  050452  020122  042502  047111
10207  050460  020107  047515  044504
10208  050466  054506  042105  053440
10209  050474  044510  044103  041440
10210  050502  052501  042523  020104
10211  050510  051105  047522  000122
10212  050516  042504  044526  042503  EM61:   .ASCIZ  /DEVICE NOT AVALABLE BEFORE COMMAND WAS TO BE GIVIEN/
10213  050524  047040  052117  040440
10214  050532  040526  040514  046102
10215  050540  020105  042502  047506
10216  050546  042522  041440  046517
10217  050554  040515  042116  053440
10218  050562  051501  052040  020117
10219  050570  042502  043440  053111
10220  050576  042511  000116
10221  050602  044122  051504  020061  EM63:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
10222  050610  047503  052116  047105
10223  050616  051524  042040  051125
10224  050624  047111  020107  047503
10225  050632  046515  047101  020104
10226  050640  040527  020123  047111
10227  050646  042440  051122  051117
10228  050654     000
10229  050655     122  041505  046101  EM64:   .ASCII  /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
10230  050662  041111  040522  042524
10231  050670  041440  046517  040515
10232  050676  042116  041440  052501
10233  050704  042523  020104  046511
10234  050712  051120  050117  051105
10235  050720  051040  043505  051511
```

```
10236  050726  042524  020122  044103
10237  050734  047101  042507  005015
10238  050742  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10239  050750  052101  020101  044507
10240  050756  042526  020123  044127
10241  050764  052101  051440  047510
10242  050772  046125  020104  042502
10243  051000  052040  042510  042522
10244  051006  005015
10245  051010  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
10246  051016  042105  042040  052101
10247  051024  020101  044507  042526
10248  051032  020123  042522  044507
10249  051040  052123  051105  041440
10250  051046  047117  042524  052116
10251  051054  020123  043101  042524
10252  051062  020122  047503  046515
10253  051070  047101  000104
10254  051074  047111  042524  051122  EM65:   .ASCIZ  /INTERRUPT FAILING/
10255  051102  050125  020124  040506
10256  051110  046111  047111  000107
10257
10258
10259
10260  051116  042510  042101  051105  EM66:   .ASCII  /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST/<15><12>
10261  051124  040440  042116  042040
10262  051132  052101  020101  047503
10263  051140  046515  047101  020104
10264  051146  047506  020122  042510
10265  051154  042101  051440  046105
10266  051162  041505  044524  047117
10267  051170  052040  051505  006524
10268  051176     012
10269  051177     103  052501  042523          .ASCII  /CAUSED ERROR/<15><12>
10270  051204  020104  051105  047522
10271  051212  006522     012
10272  051215     122  042110  052123          .ASCII  /RHDST GIVES WHAT TRACK WAS BEING WRITTEN OR READ/<15><12>
10273  051222  043440  053111  051505
10274  051230  053440  040510  020124
10275  051236  051124  041501  020113
10276  051244  040527  020123  042502
10277  051252  047111  020107  051127
10278  051260  052111  042524  020116
10279  051266  051117  051040  040505
10280  051274  006504     012
10281  051277     117  020116  054503          .ASCIZ  /ON CYLINDER 0, SECTOR 0/
10282  051304  044514  042116  051105
10283  051312  030040  020054  042523
10284  051320  052103  051117  030040
10285  051326     000
10286  051327     122  040505  020104  EM67:   .ASCII  /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST/<12><15>
10287  051334  042510  042101  051105
10288  051342  040440  042116  042040
10289  051350  052101  020101  051105
10290  051356  047522  020122  047111
10291  051364  044040  040505  020104
```

```
10292  051372  042523  042514  052103
10293  051400  047511  020116  042524
10294  051406  052123  006412
10295  051412  044506  051522  020124          .ASCII  /FIRST FOUR WORD NUMBERS ARE HEADER/<12><15>
10296  051420  047506  051125  053440
10297  051426  051117  020104  052516
10298  051434  041115  051105  020123
10299  051442  051101  020105  042510
10300  051450  042101  051105  006412
10301  051456  047527  042122  047040          .ASCII  /WORD NUMBERS 5 TO 260 ARE DATA WORDS/<12><15>
10302  051464  046525  042502  051522
10303  051472  032440  052040  020117
10304  051500  033062  020060  051101
10305  051506  020105  040504  040524
10306  051514  053440  051117  051504
10307  051522  006412
10308  051524  047111  042040  052101          .ASCIZ  /IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER/
10309  051532  020101  047527  042122
10310  051540  020123  044502  051524
10311  051546  032040  032454  033054
10312  051554  033454  034054  043440
10313  051562  053111  020105  051124
10314  051570  041501  020113  052516
10315  051576  041115  051105     000
10316  051603     122  040505  020104  EM70:   .ASCII  /READ HEADER AND DATA ERROR IN/<15><12>
10317  051610  042510  042101  051105
10318  051616  040440  042116  042040
10319  051624  052101  020101  051105
10320  051632  047522  020122  047111
10321  051640  005015
10322  051642  044504  043106  051105          .ASCII  /DIFFERENCE LINE TEST/<15><12>
10323  051650  047105  042503  046040
10324  051656  047111  020105  042524
10325  051664  052123  005015
10326  051670  047527  042122  047040          .ASCII  /WORD NOS 1-4 GIVE HEADER/<15><12>
10327  051676  051517  030440  032055
10328  051704  043440  053111  020105
10329  051712  042510  042101  051105
10330  051720  005015
10331                                          .ASCII
10332  051722  047527  042122  047040          .ASCIZ  /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
10333  051730  051517  032440  031055
10334  051736  030066  043440  053111
10335  051744  020105  040504  040524
10336  051752  053440  044510  044103
10337  051760  044440  020123  044124
10338  051766  020105  054503  044514
10339  051774  042116  051105  040440
10340  052002  042104  042522  051523
10341  052010     000
10342  052011     106  051117  044503  EM71:   .ASCII  /FORCING OPI BY 3 INDEX PULSES/<15><12>
10343  052016  043516  047440  044520
10344  052024  041040  020131  020063
10345  052032  047111  042504  020130
10346  052040  052520  051514  051505
10347  052046  005015
```

```
10348  052050  040503  051525  042105          .ASCII  /CAUSED IMPPOPER REGISTER CHANGE/<15><12>
10349  052056  044440  050115  047522
10350  052064  042520  020122  042522
10351  052072  044507  052123  051105
10352  052100  041440  040510  043516
10353  052106  006505     012
10354  052111     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10355  052116  040504  040524  043440
10356  052124  053111  051505  053440
10357  052132  040510  020124  044123
10358  052140  052517  042114  041040
10359  052146  020105  044124  051105
10360  052154  006505     012
10361  052157     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
10362  052164  042526  020104  040504
10363  052172  040524  043440  053111
10364  052200  051505  051040  043505
10365  052206  051511  042524  020122
10366  052214  047503  052116  047105
10367  052222  051524  040440  052106
10368  052230  051105  031440  044440
10369  052236  042116  054105  050040
10370  052244  046125  042523  000123
10371  052252  044124  051105  020105  EM72:   .ASCIZ  /THERE WAS AN ERROR AFTER A WRITE HEADER AND DATA COMMAND/
10372  052260  040527  020123  047101
10373  052266  042440  051122  051117
10374  052274  040440  052106  051105
10375  052302  040440  053440  044522
10376  052310  042524  044040  040505
10377  052316  042504  020122  047101
10378  052324  020104  040504  040524
10379  052332  041440  046517  040515
10380  052340  042116     000
10381  052343     122  040505  020104  EM73:   .ASCII  /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
10382  052350  042510  042101  051105
10383  052356  040440  042116  042040
10384  052364  052101  020101  047506
10385  052372  020122  030461  033071
10386  052400  020060  047527  042122
10387  052406  020123  005015
10388  052412  044124  052101  044440          .ASCII  /THAT IS 46 SECTORS /<15><12>
10389  052420  020123  033064  051440
10390  052426  041505  047524  051522
10391  052434  006440     012
10392  052437     124  040510  020124          .ASCIZ  /THAT IS OVER 3 INDEX PULSES CAUSED AN ERROR/
10393  052444  051511  047440  042526
10394  052452  020122  020063  047111
10395  052460  042504  020130  052520
10396  052466  051514  051505  041440
10397  052474  052501  042523  020104
10398  052502  047101  042440  051122
10399  052510  051117     000
10400  052513     122  040505  020104  EM74:   .ASCII  /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
10401  052520  042510  042101  051105
10402  052526  040440  042116  042040
10403  052534  052101  020101  047506
```

```
10404  052542  020122  030461  033071
10405  052550  020060  047527  042122
10406  052556  020123  005015
10407  052562  044124  052101  044440          .ASCII  /THAT IS 46 SECTORS, THAT IS OVER 3 INDEX /<15><12>
10408  052570  020123  033064  051440
10409  052576  041505  047524  051522
10410  052604  020054  044124  052101
10411  052612  044440  020123  053117
10412  052620  051105  031440  044440
10413  052626  042116  054105  006440
10414  052634     012
10415  052635     120  046125  042523          .ASCIZ  /PULSES CAUSED DPI TO SET/
10416  052642  020123  040503  051525
10417  052650  042105  047440  044520
10418  052656  052040  020117  042523
10419  052664  000124
10420
10421  052666  041520  020040  020040  DH1:    .ASCII  /PC      WAT     BIT     REG     REG     RHCS1/<15><12>
10422  052674  020040  040527  020124
10423  052702  020040  020040  044502
10424  052710  020124  020040  020040
10425  052716  042522  020107  020040
10426  052724  020040  042522  020107
10427  052732  020040  020040  044122
10428  052740  051503  006461     012
10429  052745     040  020040  020040          .ASCIZ  /       PC      EXPECT  ADDRESS CONTENT CONTENTS/
10430  052752  020040  050040  020103
10431  052760  020040  020040  042440
10432  052766  050130  041505  020124
10433  052774  040440  042104  042522
10434  053002  051523  041440  047117
10435  053010  042524  052116  041440
10436  053016  047117  042524  052116
10437  053024  000123
10438  053026  041520  020040  020040  DH4:    .ASCII  /PC      WAT     BIT     REG     TIME  IN/<15><12>
10439  053034  020040  040527  020124
10440  053042  020040  020040  044502
10441  053050  020124  020040  020040
10442  053056  042522  020107  020040
10443  053064  020040  044524  042515
10444  053072  020040  047111  005015
10445  053100  020040  020040  020040          .ASCIZ  /       PC      EXPECT  ADDRESS 10  MSEC/
10446  053106  020040  041520  020040
10447  053114  020040  020040  054105
10448  053122  042520  052103  020040
10449  053130  042101  051104  051505
10450  053136  020123  030061  020040
10451  053144  051515  041505     000
10452  053151     120  020103  020040  DH5:    .ASCII  /PC      REG     GOOD    RECEIVED/<15><12>
10453  053156  020040  051040  043505
10454  053164  020040  020040  043440
10455  053172  047517  020104  020040
10456  053200  051040  041505  044505
10457  053206  042526  006504     012
10458  053213     040  020040  020040          .ASCIZ  /       ADDRESS DATA    DATA/
10459  053220  020040  040440  042104
```

```
10460  053226  042522  051523  042040
10461  053234  052101  020101  020040
10462  053242  042040  052101  000101
10463  053250  041520  020040  020040  DH6:    .ASCII  /PC      REG     RECEIVED/<15><12>
10464  053256  020040  042522  020107
10465  053264  020040  020040  042522
10466  053272  042503  053111  042105
10467  053300  005015
10468  053302  020040  020040  020040          .ASCIZ  /       ADDRESS DATA/
10469  053310  020040  042101  051104
10470  053316  051505  020123  040504
10471  053324  040524     000
10472  053327     120  020103  020040  DH7:    .ASCIZ  /PC      REG     ADDRESS/
10473  053334  020040  051040  043505
10474  053342  020040  020040  040440
10475  053350  042104  042522  051523
10476  053356     000
10477  053357     120  020103  020040  DH10:   .ASCII  /PC      TEST    FAILING CONTENT CONTENT CONTENT CONTENT/<15><12>
10478  053364  020040  052040  051505
10479  053372  020124  020040  043040
10480  053400  044501  044514  043516
10481  053406  041440  047117  042524
10482  053414  052116  041440  047117
10483  053422  042524  052116  041440
10484  053430  047117  042524  052116
10485  053436  041440  047117  042524
10486  053444  052116  005015
10487  053450  020040  020040  020040          .ASCIZ  /       NO      REG.    RHCS1   RHCS2   RHDS1   RHER1/
10488  053456  020040  047516  020040
10489  053464  020040  020040  042522
10490  053472  027107  020040  020040
10491  053500  044122  051503  020061
10492  053506  020040  044122  051503
10493  053514  020062  020040  044122
10494  053522  051504  020061  020040
10495  053530  044122  051105  000061
10496  053536  041520  020040  020040  DH26:   .ASCII  /PC      CONT OF CONT OF CONT OF CONT OF CONT OF CONT OF/<15><12>
10497  053544  020040  047503  052116
10498  053552  047440  020106  047503
10499  053560  052116  047440  020106
10500  053566  047503  052116  047440
10501  053574  020106  047503  052116
10502  053602  047440  020106  047503
10503  053610  052116  047440  020106
10504  053616  047503  052116  047440
10505  053624  006506     012
10506  053627     040  020040  020040          .ASCIZ  /       RHCS1   RHCS2   RHDS1   RHER1   RHER2   RHER3/
10507  053634  020040  051040  041510
10508  053642  030523  020040  051040
10509  053650  041510  031123  020040
10510  053656  051040  042110  030523
10511  053664  020040  051040  042510
10512  053672  030522  020040  051040
10513  053700  042510  031122  020040
10514  053706  051040  042510  031522
10515  053714     000
```

```
10516  053715     120  020103  020040  DH30:   .ASCII  /PC     WORD    GOOD    BAD/<15><12>
10517  053722  020040  053440  051117
10518  053730  020104  020040  043440
10519  053736  047517  020104  020040
10520  053744  041040  042101  005015
10521  053752  020040  020040  020040          .ASCI7  /       NO      DATA    DATA/
10522  053760  020040  047040  020117
10523  053766  020040  020040  040504
10524  053774  040524  020040  020040
10525  054002  040504  040524     000
10526  054007     120  020103  020040  DH51:   .ASCII  /PC     REG     GOOD    RECVD   ILLEGL/<15><12>
10527  054014  020040  051040  043505
10528  054022  020040  020040  043440
10529  054030  047517  020104  020040
10530  054036  051040  041505  042126
10531  054044  020040  044440  046114
10532  054052  043505  006514     012
10533  054057     040  020040  020040          .ASCIZ  /       ADDRESS DATA    DATA    FUNCTN/
10534  054064  020040  040440  042104
10535  054072  042522  051523  042040
10536  054100  052101  020101  020040
10537  054106  042040  052101  020101
10538  054114  020040  043040  047125
10539  054122  052103  000116
10540  054126  041520  020040  020040  DH60:   .ASCII  /PC     REG     GOOD    RECVD   MODFING/<15><12>
10541  054134  020040  042522  020107
10542  054142  020040  020040  047507
10543  054150  042117  020040  020040
10544  054156  042522  053103  020104
10545  054164  020040  047515  043104
10546  054172  047111  006507     012
10547  054177     040  020040  020040          .ASCIZ  /       ADDPESS DATA    DATA    REG/
10548  054204  020040  040440  042104
10549  054212  042522  051523  042040
10550  054220  052101  020101  020040
10551  054226  042040  052101  020101
10552  054234  020040  051040  043505
10553  054242     000
10554  054243     120  020103  020040  DH61:   .ASCII  /PC     PC OF   RHDS1   /<15><12>
10555  054250  020040  050040  020103
10556  054256  043117  020040  051040
10557  054264  042110  030523  020040
10558  054272  006440     012
10559  054275     040  020040  020040          .ASCIZ  /       JSR     WAS     /
10560  054302  020040  045040  051123
10561  054310  020040  020040  053440
10562  054316  051501  020040  020040
10563  054324  000040
10564  054326  041520  020040  020040  DH62:   .ASCII  /PC     PC OF   RHCS1/<15><12>
10565  054334  020040  041520  047440
10566  054342  020106  020040  044122
10567  054350  051503  006461     012
10568  054355     040  020040  020040          .ASCIZ  /       JSR     WAS/
10569  054362  020040  045040  051123
10570  054370  020040  020040  053440
10571  054376  051501     000
```

```
10572  054401     040  020040  041520  DH65:   .ASCII  /   PC    TEST    CONT    CONT    CONT  /<15><12>
10573  054406  020040  020040  042524
10574  054414  052123  020040  020040
10575  054422  047503  052116  020040
10576  054430  020040  047503  052116
10577  054436  020040  020040  047503
10578  054444  052116  020040  005015
10579  054452  020040  020040  020040          .ASCIZ  /          NO     RHCS1   RHAS    RHDS1 /
10580  054460  020040  020040  047040
10581  054466  020117  020040  020040
10582  054474  044122  051503  020061
10583  054502  020040  044122  051501
10584  054510  020040  051040  042110
10585  054516  030523  000040
10586  054522  041520  020040  020040  DH66:   .ASCIZ  /PC      RHDST   RHER1   RHER2   RHER3   RHCS1   RHCS2/
10587  054530  020040  044122  051504
10588  054536  020124  020040  044122
10589  054544  051105  020061  020040
10590  054552  044122  051105  020062
10591  054560  020040  044122  051105
10592  054566  020063  020040  044122
10593  054574  051503  020061  020040
10594  054602  044122  051503  000062
10595  054610  041520  020040  020040  DH72:   .ASCIZ  /PC      RHCSI   RHCS2   RHDS1   RHDST   RHCA    RHER1   RHWC/
10596  054616  020040  044122  051503
10597  054624  020111  020040  044122
10598  054632  051503  020062  020040
10599  054640  044122  051504  020061
10600  054646  020040  044122  051504
10601  054654  020124  020040  044122
10602  054662  040503  020040  020040
10603  054670  044122  051105  020061
10604  054676  020040  044122  041527
10605  054704     000
10606
10607
10608          054706                  .EVEN
10609  054706  001116  030302  030306  DT1:    .WORD   $ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1,0
10610  054714  030304  001126  002250
10611  054722  000000
10612  054724  001116  030302  030306  DT4:    .WORD   $ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,WAITTM,0
10613  054732  030304  001126  030310
10614  054740  000000
10615  054742  001116  004464  001124  DT5:    .WORD   $ERRPC,REGADR,$GDDAT,$BDDAT,0
10616  054750  001126  000000
10617  054754  001116  004464  001126  DT6:    .WORD   $ERRPC,REGADR,$BDDAT,0
10618  054762  000000
10619  054764  001116  001172  000000  DT7:    .WORD   $ERRPC,$TMP1,0
10620  054772  001116  004470  001122  DT10:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
10621  055000  002250  002246  002272
10622  055006  002252  000000
10623  055012  001116  002250  002246  DT26:   .WORD   $ERRPC,CS1,CS2,DS1,ER1,ER2,ER3,0
10624  055020  002272  002252  002256
10625  055026  002264  000000
10626  055032  001116  004466  001124  DT30:   .WORD   $ERRPC,ERWORD,$GDDAT,$BDDAT,0
10627  055040  001126  000000
```

```
10628  055044  001116  004464  001124  DT51:   .WORD   $ERRPC,REGADR,$GDDAT,$BDDAT,ILLEGL,0
10629  055052  001126  002352  000000
10630  055060  001116  004464  001124  DT60:   .WORD   $ERRPC,REGADR,$GDDAT,$BDDAT,$BDADR,0
10631  055066  001126  001122  000000
10632  055074  001116  030114  001122  DT61:   .WORD   $ERRPC,PCJSR,$BDADR,0
10633  055102  000000
10634  055104  001116  030114  001122  DT62:   .WORD   $ERRPC,PCJSR,$BDADR,0
10635  055112  000000
10636  055114  001116  004470  002250  DT65:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
10637  055122  002266  002272  000000
10638  055130  001116  002252  002256  DT66:   .WORD   $ERRPC,ER1,ER2,ER3,CS1,CS2,0
10639  055136  002264  002250  002246
10640  055144  000000
10641  055146  001116  002250  002246  DT72:   .WORD   $ERRPC,CS1,CS2,DS1,DST,CA,ER1,WC,0
10642  055154  002272  002254  002262
10643  055162  002252  002242  000000
10644  055170     000     000     000  DF1:    .BYTE   0,0,0,0,0,0
10645  055173     000     000     000
10646  055176     000     000     000  DF4:    .BYTE   0,0,0,0,0,1
10647  055201     000     000     001
10648  055204     000     000     000  DF5:    .BYTE   0,0,0,0
10649  055207     000
10650  055210     000     000     000  DF6:    .BYTE   0,0,0
10651  055213     000     000          DF7:    .BYTE   0,0
10652  055215     000     000     000  DF10:   .BYTE   0,0,0,0,0,0,0
10653  055220     000     000     000
10654  055223     000
10655  055224     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0
10656  055227     000     000     000
10657  055232     000
10658  055233     000     000     000  DF30:   .BYTE   0,0,0,0
10659  055236     000
10660  055237     000     000     000  DF51:   .BYTE   0,0,0,0,0
10661  055242     000     000
10662  055244     000     000     000  DF60:   .BYTE   0,0,0,0,0
10663  055247     000     000
10664  055251     000     000     000  DF61:   .BYTE   0,0,0
10665  055254     000     000     000  DF62:   .BYTE   0,0,0
10666  055257     000     000     000  DF65:   .BYTE   0,0,0,0,0
10667  055262     000     000
10668  055264     000     000     000  DF66:   .BYTE   0,0,0,0,0,0,0
10669  055267     000     000     000
10670  055272     000
10671  055273     000     000     000  DF72:   .BYTE   0,0,0,0,0,0,0,0
10672  055276     000     000     000
10673  055301     000     000
10674          055304                  .EVEN
10675                                  .EVEN
10676
10677          000001                  .END
```

```
ACL    = 000040          1160#
ACU    = 100000          1120#
AOE    = 001000          1052#   4750    4922
AS       002266          1251#   8210*   9004   10636
ATA    = 100000          1040#   2973    3257    3386    3581    4054    4190    4360    4495    4757    4915    5301    5574
                         5871    6151    6614    6784    7910
ATABLE   004552          1313#   1666
ATTENT   004622          1339#   1664*   1666*   2996    3241    3400    3583    4065    4203    4373    4506    4763    4927
                         5308    5588    5875    6155    6618    6798
AT0    = 000001          1072#
AT1    = 000002          1073#
AT2    = 000004          1074#
AT3    = 000010          1075#
AT4    = 000020          1076#
AT5    = 000040          1077#
AT6    = 000100          1078#
AT7    = 000200          1079#
A16    = 000400          1015#
A17    = 001000          1016#
BA       002244          1240#   8932
BAI    = 000010           986#   6984
BASECH   032206          1496    8413#
BEGIN    004666           152    1353#   9276
BEGIN1   004636           156    1347#
BEGIN2   004652           154    1350#
BIT0   = 000001           125#    2014    2130    5854    6137
BIT00  = 000001           115#     125
BIT01  = 000002           114#     124
BIT02  = 000004           113#     123
BIT03  = 000010           112#     122
BIT04  = 000020           111#     121
BIT05  = 000040           110#     120
BIT06  = 000100           109#     119
BIT07  = 000200           108#     118
BIT08  = 000400           107#     117    8508
BIT09  = 001000           106#     116    8516    8890
BIT1   = 000002           124#    5833    6117
BIT10  = 002000           105#    5833    6117    8876
BIT11  = 004000           104#    8523
BIT12  = 010000           103#
BIT13  = 020000           102#    8883
BIT14  = 040000           101#
BIT15  = 100000           100#
BIT2   = 000004           123#    5833
BIT3   = 000010           122#
BIT4   = 000020           121#    7992
BIT5   = 000040           120#
BIT6   = 000100           119#
BIT7   = 000200           118#
BIT8   = 000400           117#
BIT9   = 001000           116#
BPTVEC= 000014            132#
CA       002262          1249#    8996   10641
CC       002304          1258#    9036
CHECK    030116          1001     7893#
CHECKC   030154          7897     7901#
```

```
CHECKT  030140          1970    2086    2198    2293    2425    2559    2618    2855    2943    3099    3185    3343    3472
                        3518    3709    3844    4007    4143    4250    4314    4452    4604    4697    4867    5034    5181
                        5246    5391    5531    5709    5737    5814    6031    6098    6241    6394    6587    6714    6907
                        6992    7083    7150    7268    7449    7575    7898#
CHREG   030732          1864    2002    2009    2118    2125    2968    2977    2986    3220    3229    3245    3252    3374
                        3381    3393    3569    3576    4049    4058    4185    4196    4355    4366    4480    4490    4499
                        4731    4740    4752    4901    4910    5280    5289    5296    5312    5560    5569    5581    5849
                        5856    5866    5879    6132    6139    6146    6159    6602    6609    6624    6635    6770    6779
                        6791    6802    81300
CLAREA  027730          1936    2055    2236    2253    2260    2479    2710    2823    2914    3066    3154    3299    3304
                        3310    3658    3676    3811    3974    4111    4281    4420    4579    4645    4664    4813    4820
                        4827    4989    4996    5156    5214    5352    5359    5492    5499    5685    5770    5775    5782
                        5989    6006    6060    6065    6205    6210    6358    6364    7124    7222    7242    7423    7532
                        7549    77930
CLDISK  030060          1660    1710    1750    1797    1918    2044    2193    2221    2385    2553    2587    2805    2910
                        3048    3283    3466    3497    3642    3807    3958    4105    4244    4276    4414    4563    4630
                        4799    4834    4972    5003    5140    5207    5347    5486    5661    5671    5960    5972    6200
                        6352    6509    6536    6692    6837    6869    6969    7069    7106    7213    7367    7405    7503
                        78730
CLR     = 000040         9880    7878
COMPAR  032070          2156    2521    2756    3019    3428    3792    3928    4955    5103    5458    5619    5917    6322
                        6482    7316    7609    83750
COMREG  031040          1876    2021    2137    2357    2503    2737    2890    3002    3134    3263    3407    3589    3772
                        3909    4074    4210    4380    4513    4776    4935    5073    5327    5437    5600    5894    6177
                        6306    6466    6654    6820    81970
CRLF    = 000200        8645    86820
CSF     = 000002        11078
CSU     = 000010        11098
CS1     = 002250        12448   6939*   6940    7022*   7023    7191    7302    8948    10609   10620   10623   10636   10638
                        10641
CS2     = 002246        12418   8940    10620   10623   10638   10641
DB      = 002240        12388
DCK     = 100000        10508
DCL     = 000100        11618
DCLEAR  002316          12728
DE1     = 000040        10308
DFF20   = 000002        10268
DFF5    = 000001        10258
DF1     055170           291     303     314    106448
DF10    055215           368     382     396     412    106528
DF26    055224           531    106558
DF30    055233           552     571     587     604     665     699     723     759     795     902     913    106588
DF4     055176           327    106468
DF5     055204           337     422     434     448     462     474     482     489     499     508     519     542     563
                         581     597     615     626     640     652     677     690     712     752     771     783     808
                         849     861     927    106488
DF51    055237           738    106608
DF6     055210           347    106508
DF60    055244           825    106628
DF61    055251           833    106648
DF62    055254           841    106658
DF65    055257           873    106668
DF66    055264           891    106688
DF7     055213           355    106518
DF72    055273           943     954     961    106718
DH1     052666           284     296     308    104218
```

DEPPVB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DH10     053357            360     374     388     404   10477#
DH26     053536            523   10496#
DH30     053715            547     569     585     602     663     697     721     757     793     900     911   10516#
DH4      053026            321   10438#
DH5      053151            332     417     429     443     457     469     480     487     497     506     517     540     561
                           579     595     613     624     638     650     675     688     710     750     769     781     806
                           847     859     922   10452#
DH51     054007            732   10526#
DH6      053250            343   10463#
DH60     054126            819   10540#
DH61     054243            829   10554#
DH62     054326            837   10564#
DH65     054401            867   10572#
DH66     054522            883   10586#
DH7      053327            352   10472#
DH72     054610            934     952     959   10595#
DIGB   = 000004           1027#
DISPLA= 177570             47#   8537#   8875#
DLT    = 100000            998#
DL64   = 000020           1029#
DMD    = 000001           1062#   6737    6807
DPR    = 000400           1033#   1853    2333    2630    3556    3748    3883    6280    6438    7911
DRY    = 000200           1032#   1853    2438    2580    3492    3540    4271    5762    7101    7398    7526    7911
DST      002254           1246#   8988   10641
DS1      002272           1253#   7189    7300    8956   10620   10623   10636   10641
DT       002274           1254#   1691#   9060
DTE    = 010000           1055#
DTSY   = 001000           1068#
DT1      054706            290     302     313   10609#
DT10     054772            367     381     395     411   10620#
DT26     055012            530   10623#
DT30     055032            551     570     586     603     664     698     722     758     794     901     912   10626#
DT4      054724            326   10612#
DT5      054742            336     421     433     447     461     473     481     488     498     507     518     541     562
                           500     596     614     625     639     651     676     689     711     751     770     792     807
                           848     860     926   10615#
DT51     055044            737   10628#
DT6      054754            346   10617#
DT60     055060            824   10630#
DT61     055074            832   10632#
DT62     055104            840   10634#
DT65     055114            872   10636#
DT66     055130            890   10638#
DT7      054764            354   10619#
DT72     055146            942     953     960   10641#
DVA    = 004000           1018#   1730    1844    2324    2629    3547    3739    3874    6271    6429    7903
ECH    = 000100           1049#
ECI    = 004000           1133#   1955    2073    2279    2399    2400    2401    2601    2602    2603    2842    2931    3085
                           3172    3327    3328    3329    3695    3829    3830    3831    3993    4129    4299    4300    4301
                           4437    4438    4439    4598    4683    4851    4852    4853    5020    5021    5022    5175    5232
                           5376    5377    5378    5516    5517    5518    5703    5798    5799    5800    6025    6003    6226
                           6227    6228    6380    6381    6382    6901    6985    6986    6987    7144    7261    7262    7263
                           7443    7560    7569    7570
EC1      002300           1256#   9044
EC2      002302           1257#   9052
EMTVEC= 000030             135#   1366#   1367#   1473#   1500#
```

```
EM1     036452            282    9293#
EM10    037202            350    9353#
EM11    037251            371    9360#
EM12    037356            385    9372#
EM13    037464            399    9384#
EM14    037603            415    9398#
EM15    037655            425    9406#
EM16    040016            437    9424#
EM17    040144            451    9440#
EM2     036501            294    9297#
EM20    040301            465    9457#
EM21    040450            478    9475#
EM22    040523            485    9483#
EM23    040576            492    9491#
EM24    040745            502    9509#
EM25    041110            511    9527#
EM26    041272            522    9547#
EM27    041362            534    9557#
EM3     036570            306    9307#
EM30    041600            545    9583#
EM31    041660            555    9591#
EM32    042075            566    9616#
EM33    042202            574    9628#
EM34    042404            584    9652#
EM35    042430            590    9656#
EM36    042646            600    9681#
EM37    042723            607    9689#
EM4     036651            317    9316#
EM40    043140            618    9714#
EM41    043347            629    9738#
EM42    043630            643    9770#
EM43    044106            655    9800#
EM44    044444            668    9839#
EM45    044736            680    9872#
EM46    045232            693    9905#
EM47    045345            702    9919#
EM5     036760            330    9328#
EM50    045627            715    9951#
EM51    046046            726    9977#
EM52    046313            742   10006#
EM53    046564            755   10036#
EM54    046652            762   10045#
EM55    047117            774   10075#
EM56    047405            786   10108#
EM57    047615            798   10132#
EM6     037032            340    9335#
EM60    050107            811   10166#
EM61    050516            828     836   10212#
EM63    050602            845   10221#
EM64    050655            853   10229#
EM65    051074            866   10254#
EM66    051116            877   10260#
EM67    051327            893   10286#
EM7     037120            350    9344#
EM70    051603            904   10316#
EM71    052011            916   10342#
```

DERPVB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM72    052252            930   10371#
EM73    052343            950   10381#
EM74    052513            957   10400#
ERFLG$  004616          13338#   6719*  8869*
ERR   = 040000          10398#   2975   3259   3388   4056   4192   4362   4497   4759   4917   5303   5576   5873
                         6153#   6616   6786   7189   7300
ERRVEC= 000004           1288#   1475*  1482*  1502*  8499   8500*  8502*  8505*
ERWORD  004466          12968#   8388*  8394*  10626
ER1     002252          12458#   7032*  7033   8964   10620  10623  10638  10641
ER2     002256          12478#   8972   10623  10638
ER3     002264          12508#   8980   10623  10638
EXT1  = 000001          10878#
EXT10 = 000010          10908#
EXT2  = 000002          10888#
EXT20 = 000020          10918#
EXT4  = 000004          10898#
EXT40 = 000040          10928#
FEN   = 000200          11138#
FER   = 000020          10478#   5317   5586
FILL    030776          81608#
FILLRE  027762           1995    1998   2015   2111   2114   2131   2345   2348   2351   2486   2489   2492   2495
                         2717    2720   2723   2726   2729   2879   2882   2885   2992   3123   3126   3129   3213
                         3216    3237   3367   3370   3389   3760   3763   3766   3895   3898   3901   4724   4727
                         4748    4766   4769   4894   4897   4920   4923   5061   5064   5067   5273   5276   5304
                         5418    5421   5424   5577   5862   5885   5888   6165   6168   6171   6292   6296   6299
                         6450    6453   6456   6459   6644   6648   6787  78218#
FINACC  004550          13088#   7967*
FINALA  004546          13078#   7968*
FIRST   004620          13348#   1386   1419*
FLHEAD  027704           1925    2051   2227   2244   2472   2703   2812   3055   3290   3649   3669   3965   4578
                         4636    4655   4804   4982   5148   5676   5980   5997   6879   7114   7232   7413   7541
                        77648#
FMT22 = 010000          11348#   1954   1955   1956   2072   2073   2074   2278   2279   2280   2399   2400   2401
                         2601    2602   2603   2841   2842   2843   2930   2931   2932   3084   3085   3086   3171
                         3172    3173   3327   3328   3329   3694   3695   3696   3829   3830   3831   3992   3993
                         3994    4128   4129   4130   4299   4300   4301   4437   4597   4598   4599   4682   4683
                         4684    4851   4852   4853   5020   5021   5022   5174   5175   5176   5231   5376   5377
                         5378    5516   5702   5703   5704   5798   5799   5800   6024   6025   6026   6082   6083
                         6084    6226   6227   6228   6380   6381   6382   6703   6720   6900   6901   6902   6985
                         6986    6987   7143   7144   7145   7261   7262   7263   7442   7443   7444   7568   7569
                         7570    7633
FUTABL  002310          12688#   6516
GNS   = ****** U         148    1393   1397   1401   1405   1409   1413   1417   1452   1490   1494   1542   1565
                         1569    1573   1577   1605   1611   1617   1669   1676   1683   7670   7676   8264   8270
                         8277    8281   8285   8292   8296   8416   8422   8433   8439   8445   8449   8455   8461
                         8468    8922   8930   8938   8946   8954   8962   8970   8978   8986   8994   9002   9010
                         9018    9026   9034   9042   9050   9058   9066   9230   9231   9232   9233   9234   9235
                         9236    9237   9238
GO    = 000001          10128#   1825   1980   2096   2208   2305   2324   2418   2429   2572   2629   2635   2664
                         2865    2953   3109   3197   3353   3482   3530   3547   3721   3739   3856   3874   4019
                         4155    4263   4325   4464   4616   4709   4879   5046   5193   5258   5403   5543   5721
                         5753    5826   6043   6110   6253   6271   6407   6429   6590   6742   6919   7084   7093
                         7162    7280   7390   7461   7517   7591   7992
GPV   = 000010          10288#
HCE   = 000200          10508#   2991   3250   3379
HCI   = 002000          11328#   1956   2074   2280   2401   2603   2843   2932   3086   3173   3329   3696   3831
```

```
                    3994    4130    4301    4439    4509    4684    4853    5022    5176    5233    5378    5518    5704
                    5800    6026    6084    6228    6382    6902    6987    7145    7263    7444    7570
HCRC   = 000400      1051#
HT     = 000011      8643    8683#
HT.    = 015440      1835#   1840    1987#   1992    2104#   2109    2214#   2219    2315#   2320    2436#   2441    2463#
                    2468    2578#   2583    2669#   2674    2694#   2699    2872#   2877    2960#   2965    3116#   3121
                    3205#   3210    3360#   3365    3490#   3495    3538#   3543    3730#   3735    3865#   3870    4026#
                    4031    4162#   4167    4269#   4274    4332#   4337    4471#   4476    4623#   4628    4716#   4721
                    4886#   4891    5053#   5058    5200#   5205    5265#   5270    5410#   5415    5550#   5555    5728#
                    5733    5760#   5765    5841#   5846    6050#   6055    6124#   6129    6262#   6267    6420#   6425
                    6593#   6598    6748#   6753    6929#   6934    7013#   7018    7099#   7104    7176#   7181    7287#
                    7292    7396#   7401    7473#   7478    7524#   7529    7598#   7603
IAE    = 002000      1053#   4028    4063    4164    4201    4334    4371    4473    4504
IF     = 000100      1013#   1717    1757    1980    2096    2208    2305    2324    2418    2429    2572    2629    2635
                    2664    2865    2953    3109    3197    3353    3482    3530    3547    3721    3739    3856    3874
                    4019    4155    4263    4325    4464    4616    4709    4879    5046    5193    5258    5403    5543
                    5721    5753    5826    6043    6110    6253    6271    6407    6429    6590    6919    7004    7093
                    7162    7280    7390    7461    7517    7591    8018    8030    8098    8111
ILF    = 000001      1043#   6607
ILLEGL   002352      1286#   6528*   6542    6553    6561    6589    6630    10628
ILP    = 000002      1044#
IOTVEC= 000020       133#    1364*   1365*
IP     = 000100      989#    3227    4747    5319    5320    5428    5429    5592    5593    6559    6570    6642    6814
                    6815
IXE    = 004000      1117#
KIPAR0= 172340       191#
KIPAR1= 172342       192#
KIPAR2= 172344       193#
KIPAR3= 172346       194#
KIPAR4= 172350       195#
KIPAR5= 172352       196#
KIPAR6= 172354       197#
KIPAR7= 172356       198#
KIPDR0= 172300       180#
KIPDR1= 172302       181#
KIPDR2= 172304       182#
KIPDR3= 172306       183#
KIPDR4= 172310       184#
KIPDR5= 172312       185#
KIPDR6= 172314       186#
KIPDR7= 172316       187#
LA       002306      1259#   9028
LBT    = 002000      1035#   1989    2007    2123    4761    4919    7910
LT.    = 000000      1835#   1840    1987#   1992    2104#   2109    2214#   2219    2315#   2320    2436#   2441    2463#
                    2468    2578#   2583    2669#   2674    2694#   2699    2872#   2877    2960#   2965    3116#   3121
                    3205#   3210    3360#   3365    3490#   3495    3538#   3543    3730#   3735    3865#   3870    4026#
                    4031    4162#   4167    4269#   4274    4332#   4337    4471#   4476    4623#   4628    4716#   4721
                    4886#   4891    5053#   5058    5200#   5205    5265#   5270    5410#   5415    5550#   5555    5728#
                    5733    5760#   5765    5841#   5846    6050#   6055    6124#   6129    6262#   6267    6420#   6425
                    6593#   6598    6748#   6753    6929#   6934    7013#   7018    7099#   7104    7176#   7181    7287#
                    7292    7396#   7401    7473#   7478    7524#   7529    7598#   7603
MCLK   = 000002      1063#
MCPE   = 020000      1019#
MMS    = 001000      1115#
MINX   = 000004      1064#   6738    6739    6756    6757    6760    6761    6764    6765
MMVEC  = 000250       169#
```

DERPVB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MOL    = 010000          1037#   1853    2333    2630    3556    3748    3883    6280    6438    7911
MPE    = 000400           991#
MR       002270          1252#   9020
MPD    = 000020          1066#
MSE    = 000020          1110#
MSTCK  = 000010          1065#
MWP    = 000040          1067#
MXF    = 001000           992#   6558    6568    6577    6640
NED    = 010000           995#
NEM    = 004000           994#
NHS    = 002000          1116#
NOPERA   002310          1269#
NOPUSH   004610          1326#   1347*   1351*   1354*
NOUNIT   004604          1322#   1581*   1592*   1624*   1629    7691*
NUNIT    004606          1324#   1629*   1630*
OCYL   = 100000          1163#
OF       002260          1248#   9012
OFREV  = 000200          1131#
OFSET    030044          7868#
OFSETC   002342          1282#   7869
OFSTVL   004474          1303#
OF100  = 000004          1126#
OF200  = 000010          1127#
OF25   = 000001          1124#
OF400  = 000020          1128#
OF50   = 000002          1125#
OF800  = 000040          1129#
OPERSE   031176          8259#   8726
OPI    = 020000          1056#   6796    7033
OP     = 000200           990#   3225    4745    5294    5319    5320    5428    5429    5592    5593    6558    6568    6750
                         6814    6815
PAR    = 000010          1046#
PC     =%000007            59#   1385*   1537*   1660*   1710*   1750*   1791*   1797*   1801*   1885*   1918*   1970*   2030*
                         2044*   2086*   2146*   2165*   2193*   2198*   2221*   2293*   2366*   2385*   2425*   2511*   2530*
                         2553*   2559*   2587*   2618*   2746*   2765*   2805*   2855*   2898*   2910*   2943*   3010*   3027*
                         3048*   3099*   3142*   3185*   3271*   3283*   3343*   3416*   3436*   3466*   3472*   3497*   3518*
                         3599*   3642*   3709*   3781*   3801*   3807*   3844*   3918*   3937*   3958*   4007*   4083*   4105*
                         4143*   4219*   4244*   4250*   4276*   4314*   4389*   4414*   4452*   4522*   4563*   4604*   4630*
                         4697*   4785*   4799*   4834*   4867*   4944*   4964*   4972*   5003*   5034*   5082*   5112*   5140*
                         5181*   5207*   5246*   5336*   5347*   5391*   5446*   5467*   5486*   5531*   5609*   5628*   5661*
                         5671*   5709*   5737*   5814*   5908*   5926*   5960*   5972*   6031*   6098*   6191*   6200*   6241*
                         6315*   6331*   6352*   6394*   6475*   6491*   6509*   6536*   6587*   6663*   6692*   6714*   6829*
                         6837*   6869*   6907*   6942*   6969*   6992*   7025*   7035*   7069*   7083*   7106*   7150*   7187*
                         7213*   7268*   7298*   7326*   7367*   7405*   7449*   7503*   7575*   7618*   7715*   7718*   7732*
                         7881*   7917*   8209*   8219*   8330*   8395*   8650*   8657*   8663*   8677*   8679*   8680*   8711*
                         8725*   8885*   9122*   9275
PCJSR    030114          7888#   7893*   7894*   7898*   7899*  10632   10634
PCLBUF   002234          1229#   7991*
PCLCSR   002232          1228#   1434    7965*   7992*
PCLCTR   002236          1230#   7966
PGE    = 002000           993#
PIP    = 020000          1038#
PIRQ   = 177772            45#
PIRQVE= 000240            139#
PKACK    002346          1284#   1803    1824    1843
PLU    = 020000          1119#
```

DERPVB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
PRE    = 000020          1159#
PRITEM  034666           1442*  7690*  8261*  8910#  9071*  9088   9092*
PPOG   = 001000          1034#
PP0    = 000000            62#
PP1    = 000040            63#
PP2    = 000100            64#
PP3    = 000140            65#
PP4    = 000200            66#
PP5    = 000240            67#
PP6    = 000300            68#
PP7    = 000340            69#
PS     = 177776            42#    43   1358*  1379*  1716*  1756*  1792*  2443*  2677*  7667*  8260*  8750*
PSEL   = 000000          1017#
PSU    = 000001          1156#
PSW    = 177776            43#
PUTREG  031764           6942   7025   7035   7187   7298   8209   8317#
PWRVEC= 000024            134#  1370*  1371*  9247*  9248*  9256*  9271*  9272*
RDCHR  = 104412          8771   9235#
RDLIN  = 104414          8816   9236#
RDOCT  = 104416          1454   8287   8298   8424   8441   9237#
RDY    = 000200          1014#  1717   1730   1757   1844   2106   2216   2317   2465   2629   2671   2696   2874
                         2962   3118   3207   3362   3547   3732   3867   4625   4718   4888   5055   5202   5267
                         5412   5552   5730   5843   6052   6126   6264   6422   6595   6931   7015   7170   7289
                         7475   7600   7903
READAT  002334           1279#  2075   2095   2933   2952   4440   4461   5379   5402   5519   5542   5801   5825
                         6229   6252   6270   6383   6406   6428
READIN  002350           1285#
RECALI  002314           1271#  2207   3481   7092   7389   7516
REFOR   002336           1280#  2417   2663   3330   3352   3832   3855   3873   4302   4324   4854   4878   5023
                         5045   6988   7003   7264   7279   7571   7590
REGADR  004464           1295#  1519*  1850*  1859*  2330*  2339*  2643*  2657*  3553*  3562*  3745*  3754*  3880*
                         3889*  6277*  6286*  6435*  6444*  8218*  10615  10617  10628  10630
REGSAV  034524           8868#
REGSA1  034532           1473   8870#
REINTO  003420           1290#  1937   2068   2116   2158   2261   2395   2491   2523   2597   2722   2758   2824
                         2926   3021   3067   3311   3323   3372   3430   3678   3677   3793   3812   3825   3900
                         3930   4282   4295   4421   4433   4828   4847   4896   4957   4997   5016   5066   5105
                         5360   5372   5420   5460   5500   5512   5621   5783   5794   5887   5919   6222   6298
                         6324   6365   6376   6455   6484   6981   7223   7257   7318   7533   7564   7611
RELEAS  002320           1273#
RESVEC= 000010            129#
RETCL   002344           1283#
RHAS    002210           1212#  1511   1544   6415*
RHBA    002166           1200#  1999   2115   2349   2490   2721   2883   3127   3217   3371   3764   3899   4042
                         4178   4348   4725   4895   5065   5277   5419   5886   6169   6297   6454   6541*  6564*
                         6573*  6645   6701*  6812   8347*
RHCA    002204           1210#  2010   2126   4767   4924   6721*  7841*  7859*  8344*
RHCC    002226           1219#  2730   4770   5850   6133   6460   7967   8323
RHCS1   002172           1205#  1803*  1827*  1982*  2098*  2105   2210*  2307*  2316   2431*  2456*  2464   2493
                         2574*  2637*  2695   2724   2867*  2873   2955*  2961   2978   3111*  3117   3199*  3206
                         3230   3355*  3361   3394   3484*  3532*  3570   3723*  3731   3858*  3866   4021*  4046
                         4157*  4182   4265*  4327*  4352   4466*  4481   4618*  4624   4711*  4717   4732   4881*
                         4887   4902   5048*  5054   5195*  5201   5260*  5266   5281   5405*  5411   5545*  5551
                         5561   5665   5723*  5729   5755*  5828*  5842   5857   5965   6045*  6051   6112*  6125
                         6140   6255*  6263   6409*  6421   6594   6625   6742*  6771   6921*  6930   6939   7006*
                         7014   7022   7095*  7164*  7177   7282*  7288   7392*  7463*  7474   7519*  7593*  7599
```

```
                    7843*   7860*   7869*   7873    8018    8030    8098    8111    8353*   8418    8451
RHCS2   002170      1201#   1545    1598*   3221    4044    4180    4350    4741    5290    5318    5427    5591    6636
                    6749    6813    7874    8350*
RHDB    002162      1198#   1477    8425
RHDST   002176      1207#   2016    2132    2352    2447*   2496    2678*   2727    2886    2993    3130    3238    3390
                    3767    3902    4068    4193    4363    4488    4772    4930    5068    5305    5425    5578    5863
                    6166    6300    6457    6712*   6788    7842*   8345*
RHDS1   002214      1214#   1836    1865    1988    2003    2119    2215    2437    2579    2670    2969    3253    3382
                    3491    3539    3577    4050    4186    4270    4356    4491    4753    4911    5297    5570    5761
                    5867    6147    6610    6780    7100    7397    7525    7875
RHDT    002216      1215#   1599    1601    1613    1685    1689    1691
RHEC1   002222      1217#
RHEC2   002224      1218#
RHER1   002174      1206#   1548    2987    3246    3375    4027    4059    4163    4197    4333    4367    4472    4500
                    4749    4921    5313    5582    5880    6160    6603    6792    7032    7876
RHER2   002200      1208#
RHER3   002206      1211#
RHLA    002230      1220#   7968
RHMP    002212      1213#   6736    6803
RHOF    002202      1209#   6720*   7868*   8351*
RHSN    002220      1216#   1678    1688    1690
RHWC    002164      1199#   1810    1964    1996    2081    2112    2287    2346    2409    2487    2611    2718    2850
                    2880    2938    3094    3124    3179    3214    3337    3368    3511    3703    3761    3838    3896
                    4001    4040    4137    4176    4308    4346    4446    4691    4728    4861    4898    5029    5062
                    5240    5274    5385    5422    5525    5808    5889    6092    6172    6235    6293    6388    6451
                    6540*   6563*   6572*   6581    6649    6699*   6725    6811    7826    8135    8218    8321    8323
                    8346*
RMR     = 000004    1045#   5884    6164
RPTRP1  010040      1712    1729#
RPTRP2  010136      1752    1767#
RPVEC   002160       969#   1381    1711    1751    1816*   1972*   2088*   2200*   2296*   2424*   2449*   2561*   2620*
                    2681*   2857*   2945*   3101*   3188*   3345*   3474*   3521*   3712*   3847*   4010*   4146*   4252*
                    4317*   4455*   4607*   4700*   4870*   5037*   5184*   5249*   5394*   5534*   5712*   5744*   5817*
                    6034*   6101*   6244*   6397*   6910*   6995*   7085*   7153*   7271*   7382*   7452*   7508*   7578*
                    8435    8442*   8457    8471*
RPVECT  033010      1382    8465#   8471
RP4VEC  004472      1298#   1437*   1441*   1816    1972    2088    2200    2296    2449    2561    2681    2857    2945
                    3101    3188    3345    3474    3521    3712    3847    4010    4146    4252    4317    4455    4607
                    4700    4870    5037    5184    5249    5394    5534    5712    5744    5817    6034    6101    6244
                    6397    6910    6995    7085    7153    7271    7382    7452    7508    7578
RUN     032024      1944    2063    2268    2390    2592    2831    2921    3074    3162    3317    3684    3819    3982
                    4119    4289    4428    4587    4672    4841    5010    5164    5222    5367    5507    5692    5789
                    6014    6073    6217    6371    6890    6975    7133    7251    7432    7558    83440
R0      =%000000     50#    1381*   1382*   1383*   1476*   1479*   1547*   1551*   1585*   1588*   1593*   1626*   1665*
                    1666    1711*   1712*   1713*   1751*   1752*   1753*   1809*   1831*   1846    1849    1864*   1876*
                    1925*   1936*   1944*   1963*   1995*   1998*   2002*   2009*   2015*   2021*   2051*   2055*   2063*
                    2080*   2111*   2114*   2118*   2125*   2131*   2137*   2156*   2227*   2236*   2244*   2253*   2260*
                    2268*   2286*   2310*   2326    2329    2345*   2348*   2351*   2357*   2390*   2408*   2472*   2479*
                    2486*   2489*   2492*   2495*   2503*   2521*   2568*   2592*   2610*   2622*   2627*   2629*   2639
                    2642    2703*   2710*   2717*   2720*   2723*   2726*   2729*   2737*   2756*   2812*   2823*   2831*
                    2849*   2879*   2882*   2885*   2890*   2914*   2921*   2937*   2968*   2977*   2986*   2992*   3002*
                    3019*   3055*   3066*   3074*   3093*   3123*   3126*   3129*   3134*   3154*   3162*   3178*   3213*
                    3216*   3220*   3229*   3237*   3245*   3252*   3263*   3290*   3299*   3304*   3310*   3317*   3336*
                    3367*   3370*   3374*   3381*   3389*   3393*   3407*   3428*   3487*   3503*   3510*   3534*   3549
                    3552    3569*   3576*   3589*   3649*   3658*   3669*   3676*   3684*   3702*   3726*   3741    3744
                    3760*   3763*   3766*   3772*   3792*   3811*   3819*   3837*   3861*   3876    3879    3895*   3898*
```

```
                    3901*  3909*  3928*  3965*  3974*  3982*  4000*  4049*  4058*  4074*  4111*  4119*  4136*
                    4185*  4196*  4210*  4259*  4281*  4289*  4307*  4355*  4366*  4380*  4420*  4428*  4445*
                    4480*  4490*  4499*  4513*  4570*  4579*  4587*  4636*  4645*  4655*  4664*  4672*  4690*
                    4724*  4727*  4731*  4740*  4748*  4752*  4766*  4769*  4776*  4804*  4813*  4820*  4827*
                    4841*  4860*  4894*  4897*  4901*  4910*  4920*  4923*  4935*  4955*  4982*  4989*  4996*
                    5010*  5028*  5061*  5064*  5067*  5073*  5103*  5148*  5156*  5164*  5214*  5222*  5239*
                    5273*  5276*  5280*  5289*  5296*  5304*  5312*  5327*  5352*  5359*  5367*  5384*  5418*
                    5421*  5424*  5437*  5458*  5492*  5499*  5507*  5524*  5560*  5569*  5577*  5581*  5600*
                    5619*  5676*  5685*  5692*  5741*  5770*  5775*  5782*  5789*  5807*  5832*  5833*  5835
                    5849*  5836*  5862*  5866*  5879*  5885*  5888*  5894*  5903   5904*  5905   5907*  5917*
                    5980*  5989*  5997*  6006*  6014*  6060*  6065*  6073*  6091*  6116*  6117*  6119   6132*
                    6139*  6146*  6159*  6165*  6168*  6171*  6177*  6186   6187*  6188   6190*  6205*  6210*
                    6217*  6234*  6258*  6273   6276   6292*  6296*  6299*  6306*  6322*  6358*  6364*  6371*
                    6387*  6412*  6431   6434   6450*  6453*  6456*  6459*  6466*  6482*  6516*  6518   6580*
                    6602*  6609*  6624*  6635*  6644*  6648*  6654*  6700*  6701   6703*  6705*  6706*  6707*
                    6709*  6724*  6736*  6737*  6738*  6739*  6756*  6757*  6760*  6761*  6764*  6765*  6770*
                    6779*  6787*  6791*  6802*  6820*  6879*  6890*  6924*  6975*  7009*  7114*  7124*  7133*
                    7222*  7232*  7242*  7251*  7316*  7413*  7423*  7432*  7532*  7541*  7549*  7558*  7609*
                    7693*  7695   7725*  7727   7729   7732   7767   7768   7772   7777*  7797   7798   7799
                    7806*  7824   7825   7830*  7841   7842   7845*  7859   7861*  7868   7870*  7934   7935
                    7936   7943*  7993   7997*  7998   8000   8001   8002   8003   8004   8056*  8076   8078*
                    8079   8081   8082   8083   8084   8120*  8133   8134   8136   8138   8140   8145*  8165
                    8166   8167   8168   8178*  8203   8204   8205   8206   8207*  8235*  8318   8321*  8324
                    8329*  8344   8345   8346   8347   8349   8351   8353   8354*  8381   8382   8383   8384
                    8385*  8409*  8425*  8427*  8428*  8555   8565*  8569   8585   8586   8599*  8635   8636*
                    8637   8640*  8813   8817*  8818   8821   8841*  8844*  9074   9075*  9076*  9083*  9084*
                    9085*  9086*  9087*  9088   9090   9092   9093   9098   9104   9106*  9107   9121*  9214
                    9215*  9216   9217*  9218*  9219*  9249   9270*
R1      =%000001      518  1477*  1478   1484   1511*  1512*  1513   1517   1519   1544*  1553   1556   1594
                    1717*  1730   1757*  1831   1850   2310   2330   2639   2641   2643   2687*  3487   3534
                    3553   3726   3745   3861   3880   6258   6277   6412   6435   6591*  6717*  6924   7089
                    7694*  7695   7698   7765   7767*  7772*  7776*  7794   7797*  7800*  7805*  7822   7824*
                    7826*  7827*  7829*  7873*  7880*  7901   7906   7931   7934*  7937   7942*  7994   8002*
                    8036*  8037   8046*  8047   8055*  8131   8133*  8135*  8138*  8140*  8144*  8161   8165*
                    8170*  8177*  8198   8203*  8214   8216   8221   8234*  8319   8322*  8324*  8328*  8376
                    8381*  8389   8392   8396   8408*  8426*  8429*  8556   8569*  8570   8574   8598*  8814
                    8819*  8827*  8829*  8831*  8834*  8837   8840*  9103   9104*  9109   9113   9115   9120*
                    9250   9269*
R2      =%000002      528  1478*  1545*  1546*  1550*  7766   7768*  7773*  7775*  7795   7798*  7801*  7804*
                    7823   7825*  7827   7828*  7874*  7878*  7879*  7932   7935*  7937*  7941*  7995   8003*
                    8036   8045*  8046   8054*  8132   8134*  8141*  8143*  8162   8166*  8172*  8176*  8199
                    8204*  8214   8217   8221   8233*  8320   8323*  8325*  8327*  8377   8382*  8389   8393
                    8396   8407*  8557   8568*  8572*  8575   8582*  8583*  8584   8589*  8597*  8815   8828*
                    8828*  8830*  8832*  8838   8839*  9251   9268*
R3      =%000003      538  1586*  1587*  1590*  1620*  1832   1859   2311   2339   2653   2655   2657   3488
                    3535   3562   3727   3754   3862   3889   6259   6286   6413   6444   6925   7810   7796
                    7799*  7800   7803*  7875*  7895   7900   7913   7933   7936*  7938*  7940*  7996   8007*
                    8010*  8012*  8015*  8053*  8077   8087*  8090*  8092*  8095*  8119*  8163   8167*  8170
                    8171*  8175*  8200   8205*  8226*  8232*  8378   8383*  8386   8394   8401*  8486*  8558
                    8566*  8567*  8581*  8584*  8593*  8594*  8596*  8767   8768*  8769   8772*  8773   8777
                    8779   8781*  8783*  9161   9170*  9176*  9177*  9180*  9185*  9186*  9187   9196*  9252
                    9267*
R4      =%000004      548  1548*  1549*  7876*  8164   8168*  8171   8174*  8201   8206*  8219   8231*  8379
                    8386*  8387*  8388   8405*  9162   9164*  9165*  9166*  9167   9168*  9182   9184*  9192*
                    9195*  9253   9266*
R5      =%000005      558  1591*  1598   1607   1620   1622   1625*  1832*  1855   1858   2311*  2335   2338
```

```
                    2630*  2653   2656   3488*  3535*  3558   3561   3727*  3750   3753   3862*  3885   3880
                    6259*  6282   6285   6413*  6440   6443   6517*  6522*  6708*  6710*  6925*  6949*  6950*
                    6951   6953   6954   7010*  8202   8211*  8213*  8218   8230*  8380   8384*  8395   8404*
                    8559   8561*  8563*  8570*  8574*  8589   8595*  9163   9169*  9171*  9173*  9174*  9175*
                    9176   9194*  9254   9265*
R6      =%000006      56#    58   1359*  1360*  1361
R7      =%000007      57#    59
SAVER    030220     1809   1963   2080   2286   2408   2610   2849   2937   3093   3178   3336   3510   3702
                    3837   4000   4136   4307   4445   4690   4860   5028   5239   5384   5524   5807   6091
                    6234   6387   6580   6724   7930#
SAVERE   004476     1306*  1811   1878   1965   2023   2082   2139   2288   2359   2410   2505   2612   2739
                    2851   2892   2939   2996*  3004   3095   3136   3180   3241*  3265   3338   3400*  3409
                    3512   3583*  3591   3704   3774   3839   3911   4002   4040*  4042*  4044*  4046*  4065*
                    4068*  4076   4138   4176*  4178*  4180*  4182*  4193*  4203*  4212   4309   4346*  4348*
                    4350*  4352*  4363*  4373*  4382   4447   4488*  4506*  4515   4692   4763*  4772*  4778
                    4862   4927*  4930*  4937   5030   5075   5241   5308*  5320*  5321*  5329   5386   5429*
                    5430*  5439   5526   5588*  5593*  5594*  5602   5809   5875*  5896   6093   6155*  6179
                    6236   6308   6389   6468   6582   6618*  6630*  6656   6726   6798*  6811*  6812*  6815*
                    6816*  6822   7827*  8138*  8140*  8210
SC      = 100000    1021#  2982   3234   3398   3574   4485   4736   4906   5285   5565   5861   6144   6545
                    6555   6565   6574   6629   6775   6940   7023
SC1     = 000100    1093#
SC10    = 001000    1096#
SC2     = 000200    1094#
SC20    = 002000    1097#
SC4     = 000400    1095#
SEECOM   002340     1281#  2571   4262   5752   7860
SFEKCY   030030     2568   4259   5741   7859#
SELECT   004612     1328#  1348*  1350*  1353*  1446   1632   7681
SFRCH    002322     1274#  2402   2428   2604   2627   2634   3529   3546   7843
SKI     = 040000    1162#
SN       002276     1255#  1690*  9068
SND1     005706     1389   1419#
SP      =%000006      58#  1363*  1455*  1456   1457   1472*  1483   1496*  1510*  1607*  1613*  1622*  1657*
                    1671*  1678*  1685*  1709*  1729   1749*  1767   1794*  1824*  1825*  1827   1843*  1844*
                    1845   1846   1853*  1854   1855   1915*  1979*  1980*  1982   2095*  2096*  2098   2190*
                    2207*  2208*  2210   2304*  2305*  2307   2323*  2324*  2325   2326   2333*  2334   2335
                    2417*  2418*  2419   2428*  2429*  2431   2550*  2571*  2572*  2574   2634*  2635*  2637
                    2802*  2864*  2865*  2867   2952*  2953*  2955   3045*  3108*  3109*  3111   3196*  3197*
                    3199   3352*  3353*  3355   3463*  3481*  3482*  3484   3529*  3530*  3532   3546*  3547*
                    3548   3549   3556*  3557   3558   3639*  3720*  3721*  3723   3738*  3739*  3740   3741
                    3748*  3749   3750   3855*  3856*  3858   3873*  3874*  3875   3876   3883*  3884   3885
                    3955*  4018*  4019*  4021   4102*  4154*  4155*  4157   4241*  4262*  4263*  4265   4324*
                    4325*  4327   4411*  4463*  4464*  4466   4560*  4615*  4616*  4618   4708*  4709*  4711
                    4878*  4879*  4881   5045*  5046*  5048   5137*  5192*  5193*  5195   5257*  5258*  5260
                    5318*  5319*  5321   5402*  5403*  5405   5427*  5428*  5430   5483*  5542*  5543*  5545
                    5591*  5592*  5594   5658*  5720*  5721*  5723   5752*  5753*  5755   5825*  5826*  5828
                    5903*  5907   5957*  6042*  6043*  6045   6109*  6110*  6112   6186*  6190   6252*  6253*
                    6255   6270*  6271*  6272   6273   6280*  6281   6282   6349*  6406*  6407*  6409   6420*
                    6429*  6430   6431   6438*  6439   6440   6506*  6589*  6590*  6591   6609*  6613*  6614*
                    6816   6866*  6918*  6919*  6921   7003*  7004*  7006   7066*  7092*  7093*  7095   7161*
                    7162*  7164   7279*  7280*  7282   7364*  7389*  7390*  7392   7460*  7461*  7463   7516*
                    7517*  7519   7590*  7591*  7593   7632*  7633*  7634   7672*  7678*  7686*  7722*  7765*
                    7766*  7775   7776   7794*  7795*  7796*  7803   7804   7805   7822*  7823*  7828   7829
                    7893   7895*  7896*  7898   7900*  7901*  7902*  7903   7910*  7911   7931*  7932*  7933*
                    7940   7941   7942   7993*  7994*  7995*  7996*  7997   8004*  8053   8054   8055   8056
```

```
SW2    = 000004          95#
SW3    = 000010          94#
SW4    = 000020          93#
SW5    = 000040          92#
SW6    = 000100          91#
SW7    = 000200          90#
SW8    = 000400          89#
SW9    = 001000          88#
TRITVE= 000014          130#
TDF    = 000040         1111#
TESTAD  031174          8258#
TIMCNT  030550          1718    1758    8007    8012    8073#   8087    8092
TIME1   030252          1437    7965#
TIME2   030264          1441    7967#
TKVEC  = 000060           137#  8707#   8708#
TMP0    004626          1342#   2419#   2456    2663#   2664#   2687    5665#   5832    5835#   5904    5965#   6116    6119#
                        6187
TMP1    004630          1343#   6875#   6960#
TMP4    004632          1344#
TMP5    004634          1345#   2420#   2447    2665#   2678    5666#   5920#   5966#   6334#
TN     = 000035          1657#   1658    1794#   1795    1915#   1916    2190#   2191    2550#   2551    2802#   2803    3045#
                        3046    3463#   3464    3639#   3640    3955#   3956    4102#   4103    4241#   4242    4411#   4412
                        4560#   4561    5137#   5138    5483#   5484    5658#   5659    5957#   5958    6349#   6350    6506#
                        6507    6689#   6690    6866#   6867    7066#   7067    7364#   7365
TOTALA  004624          1340#   1553#   1555#
TPVEC  = 000064           138#
TRAPVE= 000034           136#   1368#   1369#   1380#   2644#   2658#   2676#
TPE    = 040000         1020#   2984    3236    3398    4487    4738    4908    5287    5567    6555    6565    6574    6777
                        7191    7302
TRK1   = 004000         1098#
TRK10  = 040000         1101#
TRK2   = 010000         1099#
TRK20  = 100000         1102#
TRK4   = 020000         1100#
TRTVEC= 000014           131#
TSTNM   004470          1297#   1658#   1707#   1747#   1795#   1916#   2191#   2551#   2803#   3046#   3464#   3640#   3956#
                        4103#   4242#   4412#   4561#   5138#   5484#   5659#   5958#   6350#   6507#   6690#   6867#   7067#
                        7365#   8266   10620   10636
TST1    006100          1448    1462    1470#   7736
TST10   010400          1914#
TST11   011122          2189#
TST12   012060          2549#
TST13   012612          2801#
TST14   013304          3044#
TST15   014346          3462#
TST16   014664          3638#
TST17   015536          3954#
TST2    006346          1508#
TST20   016026          4101#
TST21   016276          4240#
TST22   016620          4410#
TST23   017060          4559#
TST24   020404          5136#
TST25   021306          5482#
TST26   021626          5657#
TST27   022434          5930    5956#
```

```
TST3    006414          1515    1534#
TST30   023500          6336    6348#
TST31   024036          6505#
TST32   024642          6688#
TST33   025260          6865#
TST34   025672          7065#
TST35   026462          7344    7363#
TST36   027312          7665#
TST4    007502          1561    1634    1654#   7689    7699
TST5    007752          1704#
TST6    010052          1726    1732    1744#
TST7    010142          1764    1786#
TUF   = 000100          1112#
TYPDS = 104410          1600    1623    7673    7679    7687    7723    9114    9234#
TYPE  = 104400          1391    1395    1399    1403    1407    1411    1415    1450    1488    1492    1540    1563    1567
                        1571    1575    1603    1609    1615    1621    1667    1673    1674    1680    1681    1687    7668
                        7674    7685    7688    7721    7724    8262    8268    8274    8275    8279    8283    8290    8294
                        8414    8420    8431    8437    8443    8447    8453    8459    8466    8600    8648    8724    8729
                        8775    8778    8782    8845    8847    8878    8886    8920    8928    8936    8944    8952    8960
                        8968    8976    8984    8992    9000    9008    9016    9024    9032    9040    9048    9056    9064
                        9073    9095    9097    9100    9102    9117    9188    92300   9273
TYPERR  035630          8916    9072#
TYPOC = 104402          1498    1614    1672    1679    1686    8267    8273    8419    8436    8452    8458    8470    8925
                        8933    8941    8949    8957    8965    8973    8981    8989    8997    9005    9013    9021    9029
                        9037    9045    9053    9061    9069    9081    9110    9231#
TYPON = 104406          9233#
TYPOS = 104404          9232#
UNIB  = 000020           987#
UNIT    004602          1321#   1456*   1464*   1582*   1628*   1636*   1665    1671    7672    7693    7698*   7879    8348
UNITS   004562          1320#   1580*   1586    1590    1628    7694
UNITSL  004614          1329#   1457*   1464    1636
UNLOAD  002312          1270#
UNS   = 040000          1057#
UPE   = 020000           996#
US1   = 000001           983#
US2   = 000002           984#
US4   = 000004           985#
UWR   = 000010          1150#
VUF   = 000002          1157#
VU30  = 010000          1110#
VV    = 000100          1031#   1837    1853    1869    2333    2630    3556    3748    3883    6280    6438    7896    7911
WAITBT  030306          7988#   8001*   8008    8013    8082*   8088    8093    10609   10612
WAITPC  030302          7986#   7998*   7999*   8079*   8080*   10609   10612
WAITRE  030304          7987#   8000*   8008    8013    8017    8029    8081*   8088    8093    8097    8113    10609   10612
WAITTM  030310          7966*   7989#   7990*   8037    8047    10612
WAIT.P  030312          1436    7990#
WAIT.T  030552          1440    8075#   9238
WAT   = 104420          1835    1987    2104    2214    2315    2436    2463    2578    2669    2694    2872    2960    3116
                        3205    3360    3490    3538    3730    3865    4026    4162    4269    4332    4471    4623    4716
                        4886    5053    5200    5265    5410    5550    5728    5760    5841    6050    6124    6262    6420
                        6593    6748    6929    7013    7099    7176    7287    7396    7473    7524    7598    92380
WC      002242          1239#   1879    2024    2140    2360    2506    2740    2893    3005    3137    3266    3410    3592
                        3775    3912    4077    4213    4383    4516    4779    4938    5076    5330    5440    5603    5897
                        6180    6309    6469    6657    6823    8322    8924    10641
WCE   = 040000           997#
WCF   = 000040          1048#
```

```
WCU    = 000001          1106#
WLE    = 004000          1054#
WRCHDT  002326           1276#
WRCHEK  002324           1275#
WRFROM  002354           1289#  1926   1950   2000   2052   2056   2157   2228   2237   2245   2254   2274   2350
                         2473   2480   2522   2704   2711   2757   2813   2837   2884   2915   3020   3056   3080
                         3128   3155   3167   3210   3291   3300   3305   3429   3650   3659   3690   3765   3794
                         3929   3966   3975   3988   4112   4124   4571   4580   4593   4637   4646   4656   4665
                         4670   4726   4805   4814   4821   4956   4983   4990   5104   5149   5157   5170   5215
                         5227   5278   5353   5459   5493   5620   5677   5686   5698   5771   5776   5918   5981
                         5990   5998   6007   6020   6061   6066   6078   6170   6206   6211   6323   6359   6483
                         6564   6566   6700   6880   6896   7115   7125   7139   7233   7243   7317   7414   7424
                         7438   7542   7550   7610
WPIDAT  002330           1277#  3174   3196   4131   4154   5234   5257   6085   6109
WPIFOR  002332           1278#  1957   1979   2281   2304   2323   2844   2864   3087   3108   3697   3720   3738
                         3995   4018   4600   4615   4685   4708   5177   5192   5705   5720   6027   6042   6717
                         6903   6918   7146   7161   7445   7460
WRL    = 004000          1036#
WRU    = 000400          1114#
WSU    = 000004          1108#
XE2     007216           1558   1584#
$BDADR  001122            226#  5905#  6188#  7906#  7913#  10620  10630  10632  10634
$BDDAT  001126            228#  1517#  1849#  1858#  2329#  2338#  2641#  2655#  3552#  3561#  3744#  3753#  3879#
                         3888#  6276#  6285#  6434#  6443#  8017#  8029#  8097#  8113#  8217#  8393#  10609  10612
                        10615  10617  10626  10628  10630
$BELL   001210            254#  8729   8791   8878   8898
$CHARC  033750           8664   8673#  8678#
$CMTAG  001100            214#  1358   1359   1366   1372   1373
$CM1   = 000006           240#   241#   242#   243#   244#   245#   246#
$CM2   = 000014           240#   241#   242#   243#   244#   245#   246#
$CM3   = 000006           238#   240
$CM4   = 000006           246#   247#   248#   249#   250#   251#   252#
$CNTLC  034357           8724   8791#
$CRLF   001215            256#  1621   1673   1680   1687   8274   8648   8791   8850   8886   8898   9073   9097
                         9102
$DBLK   033534           8566   8600   8608#
$DOAGN  027660           7717   7726   7736#
$DTBL   033524           8569   8604#
$ENDAD  027650            208    210   7727   7730   7732#
$ENDCT  027600           7719#
$ENDMG  027664           7685   7721   7737#
$ENULL  027701           7688   7724   7740#
$EOP    027544           1499   1579   1583   7692   7709#
$FOPCT  027572           7716#  7720
$ERFLG  001103            217#  8484   8512   8514   8520#  8541   8873#  8898
$ERMAX  001115            223#  1374#  8514   8536#  8541
$ERROR  034524           1366   1500   8865#
$ERRPC  001116            224#  8880#  8881#  8882   8898   9079   10609  10612  10615  10617  10619  10620  10623
                        10626  10628  10630  10632  10634  10636  10638  10641
$ERRTB  001220            274#  9087
$ERRTY  034670           8885   8911#
$ERTTL  001112            221#  7678   7680#  8879#  8898
$ESCAP  001206            253#  1373#  8535#  8893   8895   8898
$FILLC  001150            236#  8651   8685
$FILLS  001147            235#  8685
$GDADR  001120            225#
```

```
$GDDAT  001124          227#  1518*  1845*  1854*  2325*  2334*  2642*  2656*  3548*  3557*  3740*  3749*  3875*
                       3884*  6272*  6281*  6430*  6439*  8216*  8392*  10615  10626  10628  10630
$GET42  027622         7725#
$HD   = 000000           11
$HIOCT  034522         8838*  8849#
$ICNT   001104          218#  8527*  8528   8530*  8540
$ILLUP  036434         9247   9278#
$ITEMB  001114          222#  8882*  8898   9076
$LF     001216          257#  8782   8791   8850   8898
$LPADR  001106          219#  1375*  1656*  8289*  8300   8518*  8533*  8538   8540
$LPERR  001110          220#  1376*  6532*  7227*  7494*  8272   8299*  8518   8534*  8540   8892
$MXCNT  033316         8531   8540#
$NULL   001146          234#  8653   8685
$NWTST= 000001         1466#  1468   1504#  1506   1524#  1526   1644#  1646   1696#  1698   1738#  1740   1777#
                       1779   1902#  1904   2172#  2174   2535#  2537   2790#  2792   3030#  3032   3452#  3454
                       3608#  3610   3945#  3947   4092#  4094   4228#  4230   4401#  4403   4534#  4536   5124#
                       5126   5472#  5474   5638#  5640   5933#  5935   6340#  6342   6495#  6497   6679#  6681
                       6845#  6847   7046#  7048   7348#  7350   7656#  7658
$OCNT   036240         9160*  9189*  9202#
$OMODE  036242         9155*  9159*  9164   9167*  9178*  9204#
$OVER   033302         8495   8511   8519   8529   8537#
$PASS   001100          215#  7684*  7686   7713*  7714*  7722   7737   8525   8541
$POWER  036442         9274   9281#
$PWPAD  036430         9276#
$PWRDN  036306         1370   9247#  9271
$PWRMG  036424         9274#
$PWRUP  036354         9256   9261#
$QUES   001214          255#  8775   8791   8847   8850   8898
$RDCHP  034154         8747#  9235
$RDDEC= ****** U       9238
$RDLIN  034240         8767#  9236
$RDOCT  034364         8811#  9237
$RDSZ = 000011         8760#
$REGAD  001152          238#
$REG0   001154          240#
$REG1   001156          241#
$REG2   001160          242#
$REG3   001162          243#
$REG4   001164          244#
$REG5   001166          245#
$RESET  027646         7728   7731#
$SAVPE= ****** U       9238
$SAVR6  036440         9255*  9261   9262*  9263*  9280#
$SCOPE  033052         1364   8493#
$SETUP= 000017         1358#  1364   1366   1368   1370   1372   1373   1375   7711
$SS1  = 000000         1377#
$STUP = 177777         1358#
$SVLAD  033254         8503   8532#
$SWP  = 167700            1#     11     26     27     28     29     30     31     32    252    253    254   1372
                       1373   1375   1376   1471   1509   1535   1655   1705   1745   1787   1915   2190   2550
                       2802   3045   3463   3639   3955   4102   4241   4411   4560   5137   5483   5658   5957
                       6349   6506   6689   6866   7066   7364   7666   7706   7712   7725   7737   8485   8486
                       8487   8488   8489   8494   8506   8508   8509   8512   8513   8514   8521   8522   8523
                       8534   8537   8540   8857   8858   8859   8860   8861   8876   8883   8887   8890   8890
$SWRMK= 000000           32     33   8489   8490   8510
$TIMES  001204          252#  1372*  1471*  1509*  1535*  1655*  7666*  7712*  8521*  8528   8531*  8540
```

DERPVB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$TKB     001140            231#  8689    8709    8720
$TKCNT   033756           8689#  8704*   8727    8732*   8751    8753*
$TKINT   033776           1385   1537    1791    8704#   8725
$TKQEN=  033775           8693#  8735    8756
$TKQIN   033760           8690#  8705*   8706    8733*   8734*   8735    8737*
$TKQOU   033762           8691#  8706*   8754    8755*   8756    8758*
$TKQSR   033764           8692#  8705    8737    8758
$TKS     001136            230#  1361    8689    8710*
$TKSRV   034046           8707   8720#
$TMP0    001170            246#  1594*   1595*
$TMP1    001172            247#  1484*   1718*   1719*   1758*   1759*   6515*   6518    6520*   6524    6528    6529*   7079*
                          7204*  7218*   7341*   7402*   7486*   7495*   7624    7626*   7628    7630*   7631    7632    7635
                          7637   7639*  10619
$TMP2    001174            248#
$TMP3    001176            249#
$TMP4    001200            250#
$TMP5    001202            251#
$TN    = 000037             1#    11    1447    1461    1466    1471#   1504    1509#   1514    1524    1535#   1561    1633
                          1644   1655#   1657    1696    1705#   1706    1725    1731    1738    1745#   1746    1763    1777
                          1787#  1794    1902    1915#   2172    2190#   2535    2550#   2790    2802#   3030    3045#   3452
                          3463#  3608    3639#   3945    3955#   4092    4102#   4228    4241#   4401    4411#   4534    4560#
                          5124   5137#   5472    5483#   5638    5658#   5930    5933    5957#   6336    6340    6349#   6495
                          6506#  6679    6689#   6845    6866#   7046    7066#   7343    7348    7364#   7656    7666#
$TPB     001144            233#  8670*   8685
$TPFLG   001151            237#  8631    8685
$TPS     001142            232#  8668    8685
$TRAP    036244           1368   9214#
$TRP   = 000022           9221#  9231#   9232#   9233#   9234#   9235#   9236#   9237#   9238#   9239#
$TRPAD   036264           1436*  1440*   9210    9229#
$TSTNM   001102            216#  7683*   7711*   8484    8510    8532*   8537    8541    8875    8898
$TTYIN   034346           8768   8769    8786    8790#
$TYPBN=  ****** U         9235
$TYPDS   033320           8554#  9234
$TYPE    033544           8631#  9221    9230
$TYPEC   033704           8650   8657    8663    8668#   8669
$TYPEX   033752           8674   8676    8679#
$TYPOC   036042           9158#  9231
$TYPON   036056           9157   9160#   9233
$TYPOS   036016           9153#  9232
$XTSTR   033060           8497#
$$TRP  = 000002           9220#  9231    9232    9233    9234    9235    9236    9237    9238    9239
$0FILL   036241           9154*  9158*   9168    9203#
.      = 055304            144#   148     150#    153#    155#    201#    209#    212#    258    1289#   1290#   1306#   1320#
                          1362   1375    1376    1391    1395    1399    1402#   1403    1407    1411    1415    1450    1488
                          1491#  1492    1540    1563    1566#   1567    1570#   1571    1575    1578#   1603    1606#   1609
                          1615   1667    1670#   1674    1681    7668    7674    7677#   7737    7741    8262    8268    8275
                          8278#  8279    8283    8286#   8290    8293#   8294    8414    8420    8423#   8431    8434#   8437
                          8440#  8443    8447    8450#   8453    8459    8466    8540    8541    8608#   8685    8689    8692#
                          8693   8694#   8790#   8791    8792    8850    8898    8920    8923#   8928    8931#   8936    8944
                          8952   8960    8968    8976    8984    8992    8995#   9000    9003#   9008    9011#   9016    9019#
                          9024   9027#   9032    9035#   9040    9048    9056    9059#   9064    9067#   9124#   9258    9279
                         10608#  10674#
```

DERPVB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
ALLREG    210  8918  8927  8935  8943  8951  8959  8967  8975  8983  8991  8999  9007  9015  9023
         9031  9039  9047  9055  9063
CHANGR    210  1863  2001  2008  2117  2124  2967  2976  2985  2995  3219  3228  3240  3244  3251
         3373  3380  3392  3399  3568  3575  3582  4048  4057  4064  4184  4195  4202  4354  4365
         4372  4479  4489  4498  4505  4730  4739  4751  4762  4900  4909  4926  5279  5288  5295
         5307  5311  5559  5568  5580  5587  5848  5855  5865  5874  5878  6131  6138  6145  6154
         6158  6601  6608  6617  6622  6633  6769  6778  6790  6797  6801
CHECKD    210  1970  2086  2198  2293  2425  2559  2618  2855  2943  3099  3185  3343  3472  3518
         3709  3844  4007  4143  4250  4314  4452  4604  4697  4867  5034  5181  5246  5391  5531
         5709  5737  5814  6031  6098  6241  6394  6587  6714  6907  6992  7083  7150  7268  7449
         7575
CHECKV    210  1801
CLEARA    210  1936  2055  2236  2253  2260  2479  2710  2823  2914  3066  3154  3299  3304  3310
         3658  3676  3811  3974  4111  4281  4420  4579  4645  4664  4813  4820  4827  4989  4996
         5156  5214  5352  5359  5492  5499  5685  5770  5775  5782  5989  6006  6060  6065  6205
         6210  6358  6364  7123  7221  7241  7422  7532  7548
CMPBLK    210  2154  2519  2754  3017  3426  3790  3926  4953  5101  5456  5617  5915  6320  6480
         7314  7608
CMREGI    210  1875  2020  2136  2356  2502  2736  2889  3001  3133  3262  3406  3588  3771  3908
         4073  4209  4379  4512  4775  4934  5072  5326  5436  5599  5893  6176  6305  6465  6653
         6819
COMMEN     11   1408
DATACO    210  1943  2062  2267  2389  2591  2830  2920  3073  3161  3316  3683  3818  3981  4118
         4288  4427  4586  4671  4840  5009  5163  5221  5366  5506  5691  5788  6013  6072  6216
         6370  6888  6974  7131  7249  7430  7556
DISREG    210  5318  5427  5591
DUM       210  1803
ENDCOM     16   1408
ERROR     408  1485  1520  1723  1734  1768  1851  1860  1884  2029  2145  2164  2331  2340  2365
         2510  2529  2645  2659  2745  2764  2897  3009  3026  3141  3270  3415  3435  3554  3563
         3597  3746  3755  3780  3800  3881  3890  3917  3936  4082  4218  4388  4521  4784  4943
         4963  5081  5111  5335  5445  5466  5608  5627  5906  5925  6189  6278  6287  6314  6330
         6436  6445  6474  6490  6662  6828  6943  7026  7036  7194  7305  7325  7617  7907  7914
         8020  8022  8032  8041  8050  8100  8103  8114
ESCAPE    1408
FIHEAD    218  1924  2050  2226  2243  2471  2702  2811  3054  3289  3648  3660  3964  4569  4635
         4654  4803  4981  5147  5675  5979  5996  6877  7112  7230  7411  7539
FILLBL    218
FLSVRE    218  1995  1998  2015  2111  2114  2131  2345  2348  2351  2486  2489  2492  2495  2717
         2720  2723  2726  2729  2879  2882  2885  2992  3123  3126  3129  3213  3216  3237  3367
         3370  3389  3760  3763  3766  3895  3898  3901  4724  4727  4748  4766  4769  4894  4897
         4920  4923  5061  5064  5067  5273  5276  5304  5418  5421  5424  5577  5862  5885  5888
         6165  6168  6171  6292  6296  6299  6450  6453  6456  6459  6643  6647  6787
GOO       218  1823  1978  2094  2206  2303  2427  2570  2633  2863  2951  3107  3195  3351  3480
         3528  3719  3854  4017  4153  4261  4323  4462  4614  4707  4877  5044  5191  5256  5401
         5541  5719  5751  5824  6041  6108  6251  6405  6917  7002  7091  7160  7278  7388  7459
         7515  7589
LOAD      218
MSG      1466#  1468  1504#  1506  1523#  1526  1643#  1646  1695#  1698  1737#  1740  1776#  1779  1901#
         1904  2171#  2174  2534#  2537  2789#  2792  3030#  3032  3451#  3454  3608#  3610  3944#  3947
         4091#  4094  4227#  4230  4400#  4403  4533#  4536  5123#  5126  5472#  5474  5637#  5640  5933#
         5935  6339#  6342  6494#  6497  6678#  6681  6844#  6847  7045#  7048  7347#  7350  7656#  7658
MULT      1408
NEWTST    1408  1466  1504  1524  1644  1696  1738  1777  1902  2172  2535  2790  3030  3452  3600
         3945  4092  4228  4401  4534  5124  5472  5638  5933  6340  6495  6679  6845  7046  7340
         7656
```

```
OFFST     218
POP       1408#  5907   6190   7775   7803   7828   7940   8052   8118   8143   8174   8229   8327   8403   8595
          8839   9265
PUSH      1408#  5902   6185   7764   7793   7821   7930   7993   8075   8130   8160   8197   8317   8375   8554
          8813   9249
REORGC    218
RMCLEA    218#   1659   1796   1917   2042   2192   2220   2383   2552   2586   2804   2908   3047   3281   3465
          3496   3641   3805   3957   4104   4243   4275   4413   4562   4629   4797   4833   4970   5002   5139
          5206   5345   5485   5660   5669   5959   5970   6198   6351   6508   6534   6691   6868   6968   7068
          7105   7212   7366   7404   7501
SAVE      218#   8866
SAVTST    218#   1706   1746
SCH       218#   2621   3502
SCOPE     418#   1470   1508   1534   1654   1704   1744   1786   1914   2189   2549   2801   3044   3462   3638
          3954   4101   4240   4410   4559   5136   5482   5657   5956   6348   6505   6688   6865   7065   7363
          7665   7710
SEEKCO    218#   2567   4258   5740
SETTRA    9221#  9231   9232   9233   9234   9235   9236   9237   9238
SETUP     1408#  1358
SKIP      218#    1408#  1447   1461   1514   1633   1725   1731   1763   7343
SLASH     1408
SPACE     1408
SPECIS    218#   1809   1963   2080   2286   2408   2610   2849   2937   3093   3178   3336   3510   3702   3837
          4000   4136   4307   4445   4690   4860   5028   5239   5384   5524   5807   6091   6234   6387   6579
          6724
STARS     1408#   200    202    258    964    970   1004   1006   1179   1466   1469   1504   1507   1524   1533
          1644   1653   1696   1703   1738   1743   1777   1785   1902   1913   2172   2188   2535   2548   2790
          2800   3030   3043   3452   3461   3608   3637   3945   3953   4092   4100   4228   4239   4401   4409
          4534   4558   5124   5135   5472   5481   5639   5656   5933   5955   6340   6347   6495   6504   6679
          6687   6845   6864   7046   7064   7348   7362   7655   7656   7664   7700   8478   8541   8609   8685
          8739   8760   8796   8850   9127   9205   9242
STARTT    218#   1657   1794   1915   2190   2550   2802   3045   3463   3639   3955   4102   4241   4411   4560
          5137   5493   5658   5957   6349   6506   6689   6866   7066   7364
TJUMP     218#   1561   5930   6336
TRMTRP    9221#
TSCLR2    218#
TSCLR5    218#
TTSTNO    218#   1658   1795   1916   2191   2551   2803   3046   3464   3640   3956   4103   4242   4412   4561
          5138   5494   5659   5958   6350   6507   6690   6867   7067   7365
TYPBIN    1408#
TYPDEC    1408#  7722
TYPNUM    1408#
TYPOCS    1408#
TYPOCT    1408#  1678   1685
TYPTXT    1408#  1390   1395   1399   1403   1407   1411   1415   1450   1488   1492   1540   1562   1567   1571
          1575   1603   1609   1615   1667   1674   1681   7668   7674   8262   8268   8275   8279   8283   8290
          8294   8413   8420   8431   8437   8443   8447   8453   8459   8465   8920   8928   8936   8944   8952
          8960   8968   8976   8984   8992   9000   9008   9016   9024   9032   9040   9048   9056   9064
VFCSFT    218#   1816   1972   2088   2200   2296   2449   2561   2681   2857   2945   3101   3188   3345   3474
          3521   3712   3847   4010   4146   4252   4317   4455   4607   4700   4870   5037   5184   5249   5394
          5534   5712   5744   5817   6034   6101   6244   6397   6910   6995   7085   7153   7271   7382   7452
          7508   7578
WT        218#   1834   1986   2103   2213   2314   2435   2462   2577   2668   2693   2871   2959   3115   3204
          3359   3489   3537   3729   3864   4025   4161   4268   4331   4470   4622   4715   4885   5052   5199
          5264   5409   5549   5727   5759   5840   6049   6123   6261   6419   6592   6747   6928   7012   7098
          7175   7286   7395   7472   7523   7597
```

DERPVB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
WTT       218#  1840  1992  2109  2219  2320  2441  2468  2583  2674  2699  2877  2965  3121  3210
         3365  3495  3543  3735  3870  4031  4167  4274  4337  4476  4628  4721  4891  5058  5205
         5270  5415  5555  5733  5765  5846  6055  6129  6267  6425  6598  6753  6934  7018  7104
         7181  7292  7401  7478  7529  7603
$$CMRE    202#   240   241   242   243   244   245
$$CMTM    202#   246   247   248   249   250   251
$$ESCA    140#
$$NEWT    140#  1466  1504  1524  1644  1696  1738  1777  1902  2172  2535  2790  3030  3452  3608
         3945  4092  4228  4401  4534  5124  5472  5638  5933  6340  6495  6679  6845  7046  7348
         7656
$$SET     9221#  9231  9232  9233  9234  9235  9236  9237  9238
$$SKIP    140#
.EQUAT      1#    35
.HEADE      1#
.KT11       1#   164
.SETUP      1#  1357
.SWRHI      1#    21
.SWRLO      1#   330
.$CATC      1#   141
.$CMTA      1#   202
.$EOP       1#  7700
.$ERRO      1#  8850
.$ERRT      1#
.$POWE      1#  9242
.$RDOC      1#  8796
.$READ      1#  8685
.$SCOP      1#  8478
.$TRAP      1#  9205
.$TYPD      1#  8541
.$TYPE      1#  8609
.$TYPO      1#  9127
```

DERPVB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD     6520  6529  7200  7201  7202  7337  7338  7339  7630  7639  8045  8046  8171  8213  8288
        8574  8641  8834  9087  9156  9166
ASL     8827  8829  8831  9084  9085  9086
ASLB    8579
BCC     1597  8580
BEQ     1448  1462  1515  1557  1560  1600  1602  1634  1732  1848  1857  2328  2337  2640  2654
        3551  3560  3743  3752  3878  3887  6275  6284  6433  6442  6941  6952  7024  7034  7192
        7303  7638  7682  7692  7696  7726  7905  7912  8019  8031  8099  8112  8137  8215  8225
        8390  8400  8509  8511  8513  8517  8526  8644  8752  8822  8874  8877  8891  8894  9094
        9099  9105  9116  9183
BGE     8529  8676
BGT     7717  8588  8824  9190
BHI     8515
BHIS    6544  8040  8049
BIC     1455  5319  5320  5428  5429  5592  5593  6739  6757  6761  6765  6814  6815  7714  7902
        7910  8140  8223  8398  8427  8721  8833  8912  9180
BIS     1825  1844  1980  2096  2208  2305  2324  2418  2429  2572  2629  2635  2664  2865  2953
        2996  3109  3197  3241  3353  3400  3482  3530  3547  3583  3721  3739  3856  3874  4019
        4065  4155  4203  4263  4325  4373  4464  4506  4616  4709  4763  4879  4927  5046  5193
        5258  5308  5321  5403  5430  5543  5588  5594  5721  5753  5826  5875  6043  6110  6155
        6253  6271  6407  6429  6590  6618  6630  6738  6742  6756  6760  6764  6798  6816  6919
        7004  7093  7162  7280  7390  7461  7517  7591  7633  7868  7896  8138  8349  8428  8582
        8583  9185  9186
BISB    9076
BIT     1486  1538  1559  6524  6940  7023  7033  7189  7191  7300  7302  8008  8013  8018  8030
        8088  8093  8098  8111  8508  8516  8523  8876  8883  8890
BITB    8664
BLOS    7629  8770
BLT     8571  8587  8656  8826  9191
BMI     8495  8578
BNE     1362  1387  1480  1487  1539  1552  1589  1627  1720  1760  5929  6335  6519  6523  6525
        6554  6562  6711  6961  7190  7205  7301  7342  7487  7625  7728  7730  7774  7802  7939
        8009  8011  8014  8016  8089  8091  8094  8096  8142  8173  8227  8326  8402  8430  8524
        8576  8638  8646  8652  8665  8672  8723  8728  8736  8757  8774  8780  8884  8914  9077
        9089  9108  9181  9264
BPL     8562  8592  8632  8669  8888  9179
BR      1349  1352  1388  1392  1396  1400  1404  1408  1412  1416  1439  1451  1481  1489  1493
        1541  1564  1568  1572  1576  1604  1610  1616  1619  1668  1675  1682  1726  1764  2648
        6521  6526  6552  6560  6571  6955  7344  7627  7669  7675  7697  7897  8021  8026  8035
        8043  8102  8107  8117  8139  8263  8269  8276  8280  8284  8291  8295  8415  8421  8432
        8438  8444  8448  8454  8460  8467  8497  8503  8506  8519  8522  8573  8590  8634  8649
        8658  8667  8674  8731  8776  8835  8848  8915  8921  8929  8937  8945  8953  8961  8969
        8977  8985  8993  9001  9009  9017  9025  9033  9041  9049  9057  9065  9082  9091  9111
        9118  9157  9172  9193  9258  9279
CLR     1348  1351  1353  1354  1360  1372  1373  1518  1546  1580  1582  1591  1592  1664  6515
        6540  6541  6547  6549  6550  6551  6569  6572  6573  6575  6576  6578  6706  6707  6719
        6721  7075  7076  7077  7208  7209  7210  7378  7379  7495  7499  7500  7641  7642  7680
        7683  7711  7712  7880  7965  7990  7991  8260  8521  8535  8565  8568  8704  8749  8750
        8819  8820  9071  9075  9170  9262
CLRB    1555  6956  6958  8520  8594  8673  8781  8844
CMP     1361  1483  1599  1601  1729  1730  1767  1846  1855  2326  2335  2639  2653  3549  3550
        3741  3750  3876  3885  6273  6282  6431  6440  6518  654.  6553  6561  7628  7637  7695
        7727  7729  7903  7911  8037  8047  8083  8214  8221  8224  8389  8396  8399  8504  8520
        8586  8722  8727  8735  8756  8769  8913  9088  9090
CMPB    6951  8510  8514  8643  8645  8651  8671  8675  8773  8779  8823  8825
DEC     1479  1551  1588  1626  1630  1719  1759  5928  6334  6522  6710  6960  7204  7341  7486
```

DERPVB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        7691  7715  7773  7801  7938  8010  8015  8090  8095  8141  8172  8226  8325  8401  8429
        8753  9083
DECB    8655  9178  9199
EMT       40
HALT     148  8463  8472  8633  8889  9257  9278
INC     1550  1624  1625  6950  7483  7484  7485  7626  7684  7713  8387  8527  8572  8732  8734
        8755  8879  9184  9192  9263
INCB    6957  6959  8532  8677  8873
IOT       41
JMP      152   154   156  1389  1432  1499  1558  1561  1579  1583  5930  5931  6336  6337  6527
        7346  7636  7643  7689  7699  7736  8726  8916
JSR     1385  1537  1660  1710  1750  1791  1797  1801  1809  1864  1876  1918  1925  1936  1944
        1963  1970  1995  1998  2002  2009  2015  2021  2044  2051  2055  2063  2080  2086  2111
        2114  2118  2125  2131  2137  2156  2193  2198  2221  2227  2236  2244  2253  2260  2268
        2286  2293  2345  2348  2351  2357  2385  2390  2408  2425  2472  2479  2486  2489  2492
        2495  2503  2521  2553  2559  2568  2587  2592  2610  2618  2622  2703  2710  2717  2720
        2723  2726  2729  2737  2756  2805  2812  2823  2831  2849  2855  2879  2882  2885  2890
        2910  2914  2921  2937  2943  2968  2977  2986  2992  3002  3019  3048  3055  3066  3074
        3093  3099  3123  3126  3129  3134  3154  3162  3178  3185  3213  3216  3220  3229  3237
        3245  3252  3263  3283  3290  3299  3304  3310  3317  3336  3343  3367  3370  3374  3381
        3389  3393  3407  3428  3466  3472  3497  3503  3510  3518  3569  3576  3589  3642  3649
        3658  3669  3676  3684  3702  3709  3760  3763  3766  3772  3792  3807  3811  3819  3837
        3844  3895  3898  3901  3909  3928  3958  3965  3974  3982  4000  4007  4049  4058  4074
        4105  4111  4119  4136  4143  4185  4196  4210  4244  4250  4259  4276  4281  4289  4307
        4314  4355  4366  4380  4414  4420  4428  4445  4452  4480  4490  4499  4513  4563  4570
        4579  4587  4604  4630  4636  4645  4655  4664  4672  4690  4697  4724  4727  4731  4740
        4748  4752  4766  4769  4776  4799  4804  4813  4820  4827  4834  4841  4860  4867  4894
        4897  4901  4910  4920  4923  4935  4955  4972  4982  4989  4996  5003  5010  5020  5034
        5061  5064  5067  5073  5103  5140  5148  5156  5164  5181  5207  5214  5222  5239  5246
        5273  5276  5280  5289  5296  5304  5312  5327  5347  5352  5359  5367  5384  5391  5410
        5421  5424  5437  5458  5486  5492  5499  5507  5524  5531  5560  5569  5577  5581  5600
        5619  5661  5671  5676  5685  5692  5709  5737  5741  5770  5775  5782  5789  5807  5814
        5849  5856  5862  5866  5879  5885  5888  5894  5917  5960  5972  5980  5989  5997  6006
        6014  6031  6060  6065  6073  6091  6098  6132  6139  6146  6159  6165  6168  6171  6177
        6200  6205  6210  6217  6234  6241  6292  6296  6299  6306  6322  6352  6358  6364  6371
        6387  6394  6450  6453  6456  6459  6466  6482  6509  6536  6580  6587  6602  6609  6624
        6635  6644  6648  6654  6692  6714  6724  6770  6779  6787  6791  6802  6820  6837  6869
        6879  6890  6907  6942  6969  6975  6992  7025  7035  7069  7083  7106  7114  7124  7133
        7150  7187  7213  7222  7232  7242  7251  7268  7298  7316  7367  7405  7413  7423  7432
        7449  7503  7532  7541  7549  7558  7575  7609  7732  8209  8219  8395  8650  8657  8663
        8725  8885
MOV     1347  1350  1358  1359  1363  1364  1365  1366  1367  1368  1369  1370  1371  1375  1376
        1379  1380  1381  1382  1383  1419  1436  1437  1440  1441  1442  1456  1457  1464  1471
        1472  1473  1475  1476  1477  1478  1482  1484  1496  1500  1502  1509  1510  1511  1512
        1517  1519  1535  1544  1545  1547  1548  1549  1581  1585  1586  1587  1590  1593  1594
        1598  1607  1613  1620  1622  1628  1629  1636  1655  1656  1657  1658  1665  1671  1678
        1685  1690  1691  1707  1709  1711  1712  1713  1716  1717  1718  1747  1749  1751  1752
        1753  1756  1757  1758  1792  1794  1795  1803  1816  1824  1827  1831  1832  1843  1845
        1849  1850  1853  1854  1858  1859  1915  1916  1972  1979  1982  2088  2095  2098  2190
        2191  2200  2207  2210  2296  2304  2307  2310  2311  2323  2325  2329  2330  2333  2334
        2338  2339  2417  2419  2420  2424  2428  2431  2443  2447  2449  2456  2550  2551  2561
        2571  2574  2620  2627  2630  2634  2637  2641  2642  2643  2644  2655  2656  2657  2658
        2663  2665  2676  2677  2678  2681  2687  2802  2803  2857  2864  2867  2945  2952  2955
        3045  3046  3101  3108  3111  3188  3196  3199  3345  3352  3355  3463  3464  3474  3481
        3484  3487  3488  3521  3529  3532  3534  3535  3546  3548  3552  3553  3556  3557  3561
        3562  3639  3640  3712  3720  3723  3726  3727  3738  3740  3744  3745  3748  3749  3753
```

```
        3754    3847    3855    3858    3861    3862    3873    3875    3879    3880    3883    3884    3888    3889    3955
        3956    4010    4018    4021    4040    4042    4044    4046    4068    4102    4103    4146    4154    4157    4176
        4178    4180    4182    4193    4241    4242    4252    4262    4265    4317    4324    4327    4346    4348    4350
        4352    4363    4411    4412    4455    4463    4466    4488    4560    4561    4607    4615    4618    4700    4708
        4711    4772    4870    4878    4881    4930    5037    5045    5048    5137    5138    5184    5192    5195    5249
        5257    5260    5318    5394    5402    5405    5427    5483    5484    5534    5542    5545    5591    5658    5659
        5665    5666    5712    5720    5723    5744    5752    5755    5817    5825    5828    5832    5833    5835    5903
        5904    5905    5907    5957    5958    5965    5966    6034    6042    6045    6101    6109    6112    6116    6117
        6119    6186    6187    6188    6190    6244    6252    6255    6258    6259    6270    6272    6276    6277    6280
        6281    6285    6286    6349    6350    6397    6406    6409    6412    6413    6415    6428    6430    6434    6435
        6438    6439    6443    6444    6506    6507    6516    6517    6528    6532    6545    6555    6556    6557    6558
        6559    6563    6564    6565    6566    6567    6568    6570    6574    6577    6589    6591    6689    6690    6699
        6700    6701    6703    6705    6708    6709    6712    6717    6720    6736    6737    6811    6812    6813    6866
        6867    6873    6874    6875    6910    6918    6921    6924    6925    6939    6949    6953    6954    6995    7003
        7006    7009    7010    7022    7032    7066    7067    7079    7085    7092    7095    7153    7161    7164    7218
        7227    7271    7279    7282    7364    7365    7377    7382    7389    7392    7402    7452    7460    7463    7494
        7498    7508    7516    7519    7578    7590    7593    7631    7632    7634    7635    7640    7666    7667    7672
        7678    7686    7690    7693    7694    7698    7718    7722    7725    7765    7766    7767    7768    7772    7775
        7776    7794    7795    7796    7797    7798    7799    7800    7803    7804    7805    7822    7823    7824    7825
        7827    7828    7829    7841    7842    7843    7859    7860    7869    7873    7874    7875    7876    7878    7879
        7893    7895    7898    7900    7901    7906    7913    7931    7932    7933    7934    7935    7936    7937    7940
        7941    7942    7966    7967    7968    7992    7993    7994    7995    7996    7997    7998    8000    8001    8002
        8003    8004    8007    8012    8017    8029    8053    8054    8055    8056    8076    8077    8078    8079    8081
        8082    8084    8087    8092    8097    8110    8113    8119    8120    8131    8132    8133    8134    8143    8144
        8161    8162    8163    8164    8165    8166    8167    8168    8170    8174    8175    8176    8177    8198    8199
        8200    8201    8202    8203    8204    8205    8206    8207    8211    8216    8217    8218    8222    8230    8231
        8232    8233    8234    8261    8266    8272    8289    8299    8300    8318    8319    8320    8321    8322    8323
        8324    8327    8328    8329    8344    8345    8346    8347    8348    8350    8351    8353    8376    8377    8378
        8379    8380    8381    8382    8383    8384    8385    8386    8388    8392    8393    8397    8404    8405    8406
        8407    8408    8418    8425    8426    8435    8442    8451    8457    8471    8499    8500    8502    8505    8518
        8530    8531    8533    8534    8537    8538    8555    8556    8557    8558    8559    8560    8561    8566    8569
        8589    8595    8596    8597    8598    8599    8601    8602    8635    8636    8640    8653    8705    8706    8707
        8708    8710    8737    8747    8748    8758    8767    8768    8783    8784    8785    8786    8811    8812    8813
        8814    8815    8817    8818    8837    8838    8839    8840    8841    8869    8875    8880    8892    8895    8911
        8924    8932    8940    8948    8956    8964    8972    8980    8988    8996    9004    9012    9020    9028    9036
        9044    9052    9060    9068    9074    9079    9092    9093    9098    9103    9104    9106    9109    9113    9120
        9121    9153    9161    9162    9163    9169    9176    9194    9195    9196    9197    9198    9214    9215    9218
        9247    9248    9249    9250    9251    9252    9253    9254    9255    9256    9261    9265    9266    9267    9268
        9269    9270    9271    9272    9275
MOVB    1374    1553    1666    8210    8536    8564    8567    8581    8584    8593    8637    8662    8670    8720    8733
        8754    8772    8777    8821    8882    9154    9155    9158    9159    9160    9164    9167    9168    9187    9217
NEG     8563    9165
NOP     1433    6669    7733    7734    7735
RESET   1536    1790    7731
ROL     8494    8828    8830    8832    9171    9173    9174    9175    9177
ROR     1595
RTI     7969    8057    8121    8301    8539    8603    8642    8738    8759    8787    8842    8897    9199    9277
RTS     1885    2030    2146    2165    2366    2511    2530    2746    2765    2898    3010    3027    3142    3271    3416
        3436    3599    3781    3801    3918    3937    4083    4219    4389    4522    4785    4944    4964    5082    5112
        5336    5446    5467    5609    5628    5908    5926    6191    6315    6331    6475    6491    6663    6829    7326
        7618    7777    7806    7830    7845    7861    7870    7881    7917    7943    8145    8178    8235    8330    8354
        8409    8679    8680    8711    9122    9219
SUB     7826    7894    7899    7999    8036    8080    8135    8394    8570    8881
TRAP    9221    9231    9232    9233    9234    9235    9236    9237    9238
TST     1386    1434    1446    1632    1688    1689    7624    7681    8136    8501    8525    8575    8585    8639    8647
        8666    8709    8730    8751    8836    8843    8887    8893    9115    9182    9216
```

```
TSTB    1513  1556  8512  8577  8591  8631  8668  9107
.ASCII   255   256  9406  9424  9440  9457  9491  9509  9527  9557  9567  9591  9600  9628  9636
        9656  9665  9689  9697  9714  9722  9738  9751  9770  9783  9800  9809  9821  9824  9839
        9851  9858  9872  9883  9889  9905  9919  9930  9951  9961  9977  9987 10006 10018 10045
       10058 10075 10085 10091 10108 10119 10132 10142 10147 10166 10178 10188 10229 10238 10260
       10269 10272 10286 10295 10301 10316 10322 10326 10331 10342 10348 10354 10381 10388 10400
       10407 10421 10438 10452 10463 10477 10496 10516 10526 10540 10554 10564 10572
.ASCIZ   254   257  1394  1398  1402  1406  1410  1414  1418  1453  1491  1495  1543  1566  1570
        1574  1578  1606  1612  1618  1670  1677  1684  7671  7677  7737  8265  8271  8278  8282
        8286  8293  8297  8417  8423  8434  8440  8446  8450  8456  8462  8469  8791  8923  8931
        8939  8947  8955  8963  8971  8979  8987  8995  9003  9011  9019  9027  9035  9043  9051
        9059  9067  9123  9281  9293  9297  9307  9316  9328  9335  9344  9353  9360  9372  9384
        9398  9414  9430  9447  9463  9475  9483  9497  9515  9535  9547  9574  9583  9607  9616
        9643  9652  9672  9681  9704  9729  9758  9790  9833  9869  9896  9916  9940  9960  9994
       10025 10036 10065 10098 10130 10154 10200 10212 10221 10245 10254 10281 10308 10332 10361
       10371 10392 10415 10429 10445 10458 10468 10472 10487 10506 10521 10533 10547 10559 10568
       10579 10586 10595
.BLKB   8692  8790
.BLKW   1289  1290  1306  1320  8608
.BYTE    216   217   222   223   234   235   236   237  1313  1946  1947  2065  2066  2270  2271
        2392  2393  2594  2595  2624  2625  2833  2834  2923  2924  3076  3077  3164  3165  3319
        3320  3505  3506  3686  3687  3821  3822  3984  3985  4121  4122  4291  4292  4438  4431
        4589  4590  4674  4675  4843  4844  5012  5013  5166  5167  5224  5225  5369  5370  5509
        5510  5694  5695  5791  5792  6016  6017  6075  6076  6219  6220  6373  6374  6892  6893
        6977  6978  7135  7136  7253  7254  7434  7435  7560  7561  7740  8788  8789  9200  9201
        9202  9203 10644 10646 10648 10650 10651 10652 10655 10658 10660 10662 10664 10665 10666
       10668 10671
.ENABL     1
.END   10677
.ENDC      6    15    19    29    31    32    33    40   126   140   153   177   188   199   201
         203   211   213   238   246   252   253   254   255   258   259   965   971  1005  1007
        1180  1358  1363  1364  1366  1368  1370  1372  1373  1375  1377  1394  1398  1402  1406
        1410  1414  1418  1453  1467  1468  1469  1470  1471  1472  1491  1495  1505  1506  1507
        1508  1509  1510  1525  1526  1533  1534  1535  1536  1543  1566  1570  1574  1578  1606
        1612  1618  1645  1646  1653  1654  1655  1656  1657  1670  1677  1684  1697  1698  1703
        1704  1705  1739  1740  1743  1744  1745  1778  1779  1785  1786  1787  1827  1831  1833
        1845  1863  1864  1870  1903  1904  1913  1914  1915  1929  1930  1931  1932  1933  1950
        1954  1955  1956  1957  1980  1982  1984  1985  1993  2002  2008  2009  2015  2055  2060
        2072  2073  2074  2075  2096  2098  2100  2101  2110  2118  2124  2125  2131  2173  2174
        2188  2189  2190  2208  2210  2212  2213  2220  2231  2232  2233  2234  2248  2249  2250
        2251  2274  2278  2279  2280  2281  2305  2307  2309  2310  2312  2325  2343  2395  2399
        2400  2401  2402  2429  2431  2432  2433  2442  2469  2476  2477  2478  2479  2536  2537
        2548  2549  2550  2572  2574  2576  2577  2584  2597  2601  2602  2603  2604  2635  2637
        2638  2639  2675  2700  2707  2708  2709  2710  2791  2792  2800  2801  2802  2816  2817
        2818  2819  2820  2837  2841  2842  2843  2844  2865  2867  2869  2870  2878  2926  2930
        2931  2932  2933  2953  2955  2957  2958  2966  2968  2976  2977  2985  2986  2992  2999
        3031  3032  3043  3044  3045  3059  3060  3061  3062  3063  3064  3080  3084  3085  3086
        3087  3109  3111  3113  3114  3122  3167  3171  3172  3173  3174  3197  3199  3201  3202
        3211  3220  3228  3229  3237  3244  3245  3251  3252  3260  3294  3295  3296  3297  3298
        3299  3323  3327  3328  3329  3330  3353  3355  3357  3358  3366  3374  3380  3381  3389
        3393  3399  3403  3453  3454  3461  3462  3463  3482  3484  3486  3487  3489  3496  3530
        3532  3533  3534  3536  3548  3566  3569  3575  3576  3582  3586  3609  3610  3637  3638
        3639  3653  3654  3655  3656  3673  3674  3675  3676  3690  3694  3695  3696  3697  3721
        3723  3725  3726  3728  3740  3758  3825  3829  3830  3831  3832  3856  3858  3860  3861
        3863  3875  3893  3946  3947  3953  3954  3955  3969  3970  3971  3972  3988  3992  3993
        3994  3995  4019  4021  4023  4024  4032  4049  4057  4058  4064  4068  4093  4094  4100
```

DERPVB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        4101  4102  4124  4128  4129  4130  4131  4155  4157  4159  4160  4168  4185  4193  4196
        4202  4206  4229  4230  4239  4240  4241  4263  4265  4267  4268  4275  4295  4299  4300
        4301  4302  4325  4327  4329  4330  4338  4355  4363  4366  4372  4376  4402  4403  4409
        4410  4411  4433  4437  4438  4439  4440  4464  4466  4468  4469  4477  4480  4488  4490
        4498  4499  4505  4509  4535  4536  4558  4559  4560  4574  4575  4576  4577  4593  4597
        4598  4599  4600  4616  4618  4620  4621  4629  4640  4641  4642  4643  4659  4660  4661
        4662  4678  4682  4683  4684  4685  4709  4711  4713  4714  4722  4731  4739  4740  4748
        4752  4760  4762  4766  4808  4809  4810  4811  4847  4851  4852  4853  4854  4879  4881
        4883  4884  4892  4901  4909  4910  4918  4920  4930  4986  4987  4988  4989  5016  5020
        5021  5022  5023  5046  5048  5050  5051  5059  5125  5126  5135  5136  5137  5152  5153
        5154  5155  5170  5174  5175  5176  5177  5193  5195  5197  5198  5206  5227  5231  5232
        5233  5234  5258  5260  5262  5263  5271  5280  5288  5289  5295  5296  5304  5311  5312
        5318  5372  5376  5377  5378  5379  5403  5405  5407  5408  5416  5473  5474  5481  5482
        5483  5512  5516  5517  5518  5519  5543  5545  5547  5548  5556  5560  5568  5569  5577
        5581  5587  5591  5639  5640  5656  5657  5658  5680  5681  5682  5683  5698  5702  5703
        5704  5705  5721  5723  5725  5726  5734  5753  5755  5757  5758  5766  5794  5798  5799
        5800  5801  5826  5828  5830  5831  5847  5849  5855  5856  5862  5866  5874  5878  5879
        5885  5934  5935  5955  5956  5957  5984  5985  5986  5987  6001  6002  6003  6004  6020
        6024  6025  6026  6027  6043  6045  6047  6048  6056  6078  6082  6083  6084  6085  6110
        6112  6114  6115  6130  6132  6138  6139  6145  6146  6154  6158  6159  6165  6222  6226
        6227  6228  6229  6253  6255  6257  6258  6260  6272  6290  6341  6342  6347  6348  6349
        6376  6380  6381  6382  6383  6407  6409  6411  6412  6414  6430  6448  6496  6497  6504
        6505  6506  6599  6602  6608  6609  6617  6621  6624  6630  6635  6643  6680  6681  6687
        6688  6689  6754  6770  6778  6779  6787  6791  6797  6801  6802  6808  6846  6847  6864
        6865  6866  6883  6884  6885  6886  6896  6900  6901  6902  6903  6919  6921  6923  6924
        6926  6935  6981  6985  6986  6987  6988  7004  7006  7008  7009  7011  7019  7047  7048
        7064  7065  7066  7093  7095  7097  7098  7105  7118  7119  7120  7121  7139  7143  7144
        7145  7146  7162  7164  7166  7167  7182  7236  7237  7238  7239  7257  7261  7262  7263
        7264  7280  7282  7284  7285  7293  7349  7350  7362  7363  7364  7390  7392  7394  7395
        7402  7417  7418  7419  7420  7438  7442  7443  7444  7445  7461  7463  7465  7466  7479
        7517  7519  7521  7522  7530  7545  7546  7547  7548  7564  7568  7569  7570  7571  7591
        7593  7595  7596  7604  7656  7657  7658  7664  7665  7666  7667  7671  7677  7701  7704
        7705  7706  7708  7711  7717  7720  7721  7725  7737  7740  7741  8265  8271  8278  8282
        8286  8293  8297  8417  8423  8434  8440  8446  8450  8456  8462  8469  8479  8485  8490
        8494  8496  8507  8510  8511  8512  8514  8516  8523  8527  8532  8537  8540  8541  8542
        8610  8686  8719  8727  8740  8760  8761  8768  8770  8773  8775  8790  8791  8792  8797
        8806  8850  8851  8857  8873  8880  8886  8887  8897  8898  8923  8931  8939  8947  8955
        8963  8971  8979  8987  8995  9003  9011  9019  9027  9035  9043  9051  9059  9067  9128
        9206  9215  9218  9220  9230  9231  9232  9233  9234  9235  9236  9237  9238  9243  9255
        9265  9275  9277  9284

.EQUIV    40    41    43    58    59    88    89    90    91    92    93    94    95    96    97
         116   117   118   119   120   121   122   123   124   125

.EVEN    1394  1398  1402  1406  1410  1414  1418  1453  1491  1495  1543  1566  1570  1574  1578
         1606  1612  1618  1670  1677  1684  7671  7677  8265  8271  8278  8282  8286  8293  8297
         8417  8423  8434  8440  8446  8450  8456  8462  8469  8694  8923  8931  8939  8947  8955
         8963  8971  8979  8987  8995  9003  9011  9019  9027  9035  9043  9051  9059  9067  9124
         9283 10608 10674 10675

.IF         2    12    16    29    30    31    32    33    38    98   126   148   177   188   199
          200   202   208   212   238   246   252   253   254   258   964   970  1004  1006  1179
         1358  1363  1364  1366  1368  1370  1372  1373  1375  1393  1397  1401  1405  1409  1413
         1417  1452  1466  1468  1469  1471  1472  1490  1494  1504  1506  1507  1509  1510  1524
         1526  1533  1535  1536  1542  1565  1569  1573  1577  1605  1611  1617  1644  1646  1653
         1655  1656  1669  1676  1683  1696  1698  1703  1705  1738  1740  1743  1745  1777  1779
         1785  1787  1825  1827  1829  1831  1841  1845  1864  1870  1902  1904  1913  1915  1928
         1929  1930  1931  1932  1933  1948  1950  1953  1954  1955  1956  1980  1984  1985  1993
         2002  2008  2009  2015  2054  2055  2067  2071  2072  2073  2074  2096  2100  2101  2110
```

2118 2124 2125 2131 2172 2174 2188 2190 2208 2212 2213 2220 2230 2231 2232
2233 2234 2247 2248 2249 2250 2251 2272 2274 2277 2278 2279 2280 2305 2309
2310 2321 2325 2394 2398 2399 2400 2401 2429 2432 2433 2442 2469 2475 2476
2477 2478 2479 2535 2537 2548 2550 2572 2576 2577 2584 2596 2600 2601 2602
2603 2635 2638 2639 2675 2700 2706 2707 2708 2709 2710 2790 2792 2800 2802
2815 2816 2817 2818 2819 2820 2835 2837 2840 2841 2842 2843 2865 2869 2870
2878 2925 2929 2930 2931 2932 2953 2957 2958 2966 2968 2974 2976 2977 2983
2985 2986 2992 2996 2999 3030 3032 3043 3045 3058 3059 3060 3061 3062 3063
3064 3078 3080 3083 3084 3085 3086 3109 3113 3114 3122 3166 3170 3171 3172
3173 3197 3201 3202 3211 3220 3226 3228 3229 3235 3237 3241 3244 3245 3251
3252 3258 3260 3293 3294 3295 3296 3297 3298 3299 3321 3326 3327 3328 3329
3353 3357 3358 3366 3374 3380 3381 3387 3389 3393 3399 3400 3403 3452 3454
3461 3463 3482 3486 3487 3496 3530 3533 3534 3544 3548 3569 3575 3576 3582
3583 3586 3608 3610 3637 3639 3652 3653 3654 3655 3656 3672 3673 3674 3675
3676 3688 3690 3693 3694 3695 3696 3721 3725 3726 3736 3740 3823 3828 3829
3830 3831 3856 3860 3861 3871 3875 3945 3947 3953 3955 3966 3969 3970 3971
3972 3986 3988 3991 3992 3993 3994 4019 4023 4024 4032 4049 4055 4057 4058
4064 4065 4068 4092 4094 4100 4102 4123 4127 4128 4129 4130 4155 4159 4160
4168 4185 4191 4193 4196 4202 4203 4206 4228 4230 4239 4241 4263 4267 4268
4275 4293 4298 4299 4300 4301 4325 4329 4330 4338 4355 4361 4363 4366 4372
4373 4376 4401 4403 4409 4411 4432 4436 4437 4438 4439 4464 4468 4469 4477
4480 4486 4488 4490 4496 4498 4499 4505 4506 4509 4534 4536 4558 4560 4573
4574 4575 4576 4577 4591 4593 4596 4597 4598 4599 4616 4620 4621 4629 4639
4640 4641 4642 4643 4658 4659 4660 4661 4662 4676 4678 4681 4682 4683 4684
4709 4713 4714 4722 4731 4737 4739 4740 4746 4748 4752 4758 4760 4762 4763
4766 4807 4808 4809 4810 4811 4845 4850 4851 4852 4853 4879 4883 4884 4892
4901 4907 4909 4910 4916 4918 4920 4927 4930 4985 4986 4987 4988 4989 5014
5019 5020 5021 5022 5046 5050 5051 5059 5124 5126 5135 5137 5151 5152 5153
5154 5155 5168 5170 5173 5174 5175 5176 5193 5197 5198 5206 5226 5230 5231
5232 5233 5258 5262 5263 5271 5280 5286 5288 5289 5295 5296 5302 5304 5308
5311 5312 5318 5371 5375 5376 5377 5378 5403 5407 5408 5416 5472 5474 5481
5483 5511 5515 5516 5517 5518 5543 5547 5548 5556 5560 5566 5568 5569 5575
5577 5581 5587 5588 5591 5638 5640 5656 5658 5679 5680 5681 5682 5683 5696
5698 5701 5702 5703 5704 5721 5725 5726 5734 5753 5757 5758 5766 5793 5797
5798 5799 5800 5826 5830 5831 5847 5849 5855 5856 5862 5866 5872 5874 5875
5878 5879 5885 5933 5935 5955 5957 5983 5984 5985 5986 5987 6000 6001 6002
6003 6004 6018 6020 6023 6024 6025 6026 6043 6047 6048 6056 6077 6081 6082
6083 6084 6110 6114 6115 6130 6132 6138 6139 6145 6146 6152 6154 6155 6158
6159 6165 6221 6225 6226 6227 6228 6253 6257 6258 6268 6272 6340 6342 6347
6349 6375 6379 6380 6381 6382 6407 6411 6412 6426 6430 6495 6497 6504 6506
6599 6602 6608 6609 6615 6617 6618 6621 6624 6630 6635 6641 6643 6679 6681
6687 6689 6754 6770 6776 6778 6779 6785 6787 6791 6797 6798 6801 6802 6808
6845 6847 6864 6866 6882 6883 6884 6885 6886 6894 6896 6899 6900 6901 6902
6919 6923 6924 6935 6979 6984 6985 6986 6987 7004 7008 7009 7019 7046 7048
7064 7066 7093 7097 7098 7105 7117 7118 7119 7120 7121 7137 7139 7142 7143
7144 7145 7162 7166 7167 7182 7235 7236 7237 7238 7239 7255 7260 7261 7262
7263 7280 7284 7285 7293 7348 7350 7362 7364 7390 7394 7395 7402 7416 7417
7418 7419 7420 7436 7438 7441 7442 7443 7444 7461 7465 7466 7479 7517 7521
7522 7530 7544 7545 7546 7547 7548 7562 7567 7568 7569 7570 7591 7595 7596
7604 7655 7656 7658 7664 7666 7667 7670 7676 7700 7704 7705 7706 7707 7708
7710 7716 7719 7721 7725 7736 7737 8264 8270 8277 8281 8285 8292 8296 8416
8422 8433 8439 8445 8449 8455 8461 8468 8478 8484 8489 8494 8506 8508 8509
8510 8512 8513 8514 8523 8525 8534 8539 8540 8541 8609 8685 8689 8717 8722
8739 8760 8768 8769 8773 8774 8790 8791 8796 8802 8818 8850 8856 8866 8876
8883 8885 8886 8887 8890 8897 8898 8922 8930 8938 8946 8954 8962 8970 8978
8986 8994 9002 9010 9018 9026 9034 9042 9050 9058 9066 9127 9205 9214 9218

DERPVB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
          9220    9221    9231    9232    9233    9234    9235    9236    9237    9238    9242    9255    9265    9273    9275
          9281
.IFF        29      31      32      33      40     201     203     212     238     259     965     971    1005    1007    1180
          1363    1467    1468    1470    1471    1472    1505    1506    1508    1509    1510    1525    1526    1534    1535
          1536    1645    1646    1654    1655    1656    1697    1698    1704    1705    1739    1740    1744    1745    1778
          1779    1786    1787    1864    1903    1904    1914    1915    1948    1950    1954    1955    1956    2002    2009
          2067    2072    2073    2074    2118    2125    2173    2174    2189    2190    2272    2274    2278    2279    2280
          2394    2399    2400    2401    2536    2537    2549    2550    2596    2601    2602    2603    2791    2792    2801
          2802    2835    2837    2841    2842    2843    2925    2930    2931    2932    2960    2977    2986    2999    3031
          3032    3044    3045    3078    3080    3084    3085    3086    3166    3171    3172    3173    3220    3229    3244
          3245    3252    3321    3323    3327    3328    3329    3374    3381    3393    3403    3453    3454    3462    3463
          3569    3576    3586    3609    3610    3638    3639    3688    3690    3694    3695    3696    3823    3825    3829
          3830    3831    3946    3947    3954    3955    3986    3988    3992    3993    3994    4049    4058    4068    4093
          4094    4101    4102    4123    4128    4129    4130    4185    4196    4206    4229    4230    4240    4241    4293
          4295    4299    4300    4301    4355    4366    4376    4402    4403    4410    4411    4432    4437    4439    4480
          4490    4499    4509    4535    4536    4559    4560    4591    4593    4597    4598    4599    4676    4678    4682
          4683    4684    4731    4740    4752    4766    4845    4847    4851    4852    4853    4901    4910    4930    5014
          5016    5020    5021    5022    5125    5126    5136    5137    5168    5170    5174    5175    5176    5226    5231
          5232    5233    5280    5289    5296    5311    5312    5371    5376    5377    5378    5473    5474    5482    5483
          5511    5516    5518    5560    5569    5581    5591    5639    5640    5657    5658    5696    5698    5702    5703
          5704    5793    5798    5799    5800    5849    5856    5866    5878    5879    5934    5935    5956    5957    6018
          6020    6024    6025    6026    6077    6082    6083    6084    6132    6139    6146    6158    6159    6221    6226
          6227    6228    6341    6342    6348    6349    6375    6380    6381    6382    6496    6497    6505    6506    6602
          6609    6621    6624    6635    6680    6681    6688    6689    6770    6779    6791    6801    6802    6846    6847
          6865    6866    6894    6896    6900    6901    6902    6979    6981    6984    6986    6987    7047    7048    7065
          7066    7137    7139    7143    7144    7145    7255    7257    7261    7262    7263    7349    7350    7363    7364
          7436    7438    7442    7443    7444    7562    7564    7568    7569    7570    7656    7657    7658    7665    7666
          7667    7701    7707    7711    7717    7720    7737    8479    8507    8510    8511    8514    8540    8542    8610
          8686    8740    8743    8760    8761    8769    8774    8790    8797    8851    8856    8876    8897    8898    9120
          9206    9215    9243    9275
.IFT      1394    1398    1402    1406    1410    1414    1418    1453    1491    1495    1543    1566    1570    1574    1578
          1606    1612    1618    1670    1677    1684    1864    1948    1950    1953    1954    1955    1956    2002    2009
          2067    2071    2072    2073    2074    2118    2125    2272    2274    2277    2278    2279    2280    2394    2398
          2399    2400    2401    2596    2600    2601    2602    2603    2835    2837    2840    2841    2842    2843    2925
          2929    2930    2931    2932    2968    2977    2986    2996    3078    3080    3083    3084    3085    3086    3166
          3170    3171    3172    3173    3220    3229    3241    3245    3252    3321    3326    3327    3328    3329    3374
          3381    3393    3400    3569    3576    3583    3688    3690    3693    3694    3695    3696    3823    3828    3829
          3830    3831    3986    3988    3991    3992    3993    3994    4049    4058    4065    4123    4127    4128    4129
          4130    4185    4196    4203    4293    4298    4299    4300    4301    4355    4366    4373    4432    4436    4437
          4438    4439    4480    4490    4499    4506    4591    4593    4596    4597    4598    4599    4676    4678    4681
          4682    4683    4684    4731    4740    4752    4763    4845    4850    4851    4852    4853    4901    4910    4927
          5014    5019    5020    5021    5022    5168    5170    5173    5174    5175    5176    5226    5230    5231    5232
          5233    5280    5289    5296    5308    5312    5371    5375    5376    5377    5378    5511    5515    5516    5517
          5518    5560    5569    5581    5588    5696    5698    5701    5702    5703    5704    5793    5797    5798    5799
          5800    5849    5856    5866    5875    5879    6018    6020    6023    6024    6025    6026    6077    6081    6082
          6083    6084    6132    6139    6146    6155    6159    6221    6225    6226    6227    6228    6375    6379    6380
          6381    6382    6602    6609    6618    6624    6635    6770    6779    6791    6798    6802    6894    6896    6899
          6900    6901    6902    6979    6984    6985    6986    6987    7137    7139    7142    7143    7144    7145    7255
          7260    7261    7262    7263    7436    7438    7441    7442    7443    7444    7562    7567    7568    7569    7570
          7671    7677    8265    8271    8278    8282    8286    8293    8297    8417    8423    8434    8440    8446    8450
          8456    8462    8469    8522    8742    8747    8823    8843    8850    8886    8923    8931    8939    8947    8955
          8963    8971    8979    8987    8995    9003    9011    9019    9027    9035    9043    9051    9059    9067
.IFTF     1394    1398    1402    1406    1410    1414    1418    1453    1491    1495    1543    1566    1570    1574    1578
          1606    1612    1618    1670    1677    1684    7671    7677    8265    8271    8278    8282    8286    8293    8297
          8417    8423    8434    8440    8446    8450    8456    8462    8469    8520    8740    8743    8819    8827    8849
          8885    8923    8931    8939    8947    8955    8963    8971    8979    8987    8995    9003    9011    9019    9027
```

DERPVB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
          9035    9043    9051    9059    9067
.IIF         1       6      11      26      27      28      29      32      33      34      35     140     250    1364    1366
          1372    1373    1375    1376    1679    1686    1804    1805    1828    1829    1958    1959    1983    1984    2076
          2077    2099    2100    2211    2212    2282    2283    2308    2309    2403    2404    2432    2575    2576    2605
          2606    2638    2845    2846    2868    2869    2934    2935    2956    2957    3088    3089    3112    3113    3175
          3176    3200    3201    3331    3332    3356    3357    3485    3486    3533    3698    3699    3724    3725    3833
          3834    3859    3860    3996    3997    4022    4023    4132    4133    4158    4159    4266    4267    4303    4304
          4328    4329    4441    4442    4467    4468    4601    4602    4619    4620    4686    4687    4712    4713    4855
          4856    4882    4883    5024    5025    5049    5050    5178    5179    5196    5197    5235    5236    5261    5262
          5380    5381    5406    5407    5520    5521    5546    5547    5706    5707    5724    5725    5756    5757    5802
          5803    5829    5830    6028    6029    6046    6047    6086    6087    6113    6114    6230    6231    6256    6257
          6384    6385    6410    6411    6904    6905    6922    6923    6989    6990    7007    7008    7096    7097    7147
          7148    7165    7166    7265    7266    7283    7284    7393    7394    7446    7447    7464    7465    7520    7521
          7572    7573    7594    7595    7705    7711    7712    7723    7737    7741    8485    8486    8487    8488    8489
          8490    8521    8522    8537    8540    8541    8685    8689    8694    8783    8791    8792    8850    8857    8858
          8859    8860    8861    8898    9230    9231    9232    9233    9234    9235    9236    9237    9238
.IRP      1358    1466    1504    1524    1644    1696    1738    1777    1902    2172    2535    2790    3030    3452    3600
          3945    4092    4228    4401    4534    5124    5472    5638    5903    5907    5933    6186    6190    6340    6495
          6679    6845    7046    7348    7656    7765    7775    7794    7803    7822    7828    7931    7940    7993    8053
          8076    8119    8131    8143    8161    8174    8198    8230    8310    8327    8376    8404    8555    8595    8813
          8839    8866    9249    9265
.LIST        1      11      20      21      32     140     148     238     240     241     242     243     244     245     246
           247     248     249     250     251     252    1358    1377    1394    1398    1402    1406    1410    1414    1418
          1453    1466    1471    1491    1495    1504    1509    1524    1535    1543    1566    1570    1574    1578    1606
          1612    1618    1644    1655    1657    1670    1677    1684    1696    1705    1738    1745    1777    1787    1794
          1835    1840    1841    1902    1915    1987    1992    1993    2104    2109    2110    2172    2190    2214    2219
          2220    2315    2320    2321    2436    2441    2442    2463    2468    2469    2535    2550    2578    2583    2584
          2669    2674    2675    2694    2699    2700    2790    2802    2872    2877    2878    2960    2965    2966    3030
          3045    3116    3121    3122    3205    3210    3211    3360    3365    3366    3452    3463    3490    3495    3496
          3538    3543    3544    3608    3639    3730    3735    3736    3865    3870    3871    3945    3955    4026    4031
          4032    4092    4102    4162    4167    4168    4228    4241    4269    4274    4275    4332    4337    4338    4401
          4411    4471    4476    4477    4534    4560    4623    4628    4629    4716    4721    4722    4886    4891    4892
          5053    5058    5059    5124    5137    5200    5205    5206    5265    5270    5271    5410    5415    5416    5472
          5483    5550    5555    5556    5638    5658    5728    5733    5734    5760    5765    5766    5841    5846    5847
          5933    5957    6050    6055    6056    6124    6129    6130    6262    6267    6268    6340    6349    6420    6425
          6426    6495    6506    6593    6598    6599    6679    6689    6748    6753    6754    6845    6866    6929    6934
          6935    7013    7018    7019    7046    7066    7099    7104    7105    7176    7181    7182    7287    7292    7293
          7348    7364    7396    7401    7402    7473    7478    7479    7524    7529    7530    7598    7603    7604    7656
          7666    7671    7677    8265    8271    8278    8282    8286    8293    8297    8417    8423    8434    8440    8446
          8450    8456    8462    8469    8489    8760    8923    8931    8939    8947    8955    8963    8971    8979    8987
          8995    9003    9011    9019    9027    9035    9043    9051    9059    9067    9220    9221    9230    9231    9232
          9233    9234    9235    9236    9237    9238    9239
.MACRO       21      33     202    1466    1504    1523    1643    1695    1737    1776    1901    2171    2534    2789    3030
          3451    3608    3944    4091    4227    4400    4533    5123    5472    5637    5933    6339    6494    6678    6844
          7045    7347    7656    9221
.MCALL       1     140
.NLIST       1      11      20      21      32     140     148     238     240     241     242     243     244     245     246
           247     248     249     250     251     252    1358    1377    1394    1398    1402    1406    1410    1414    1418
          1453    1466    1471    1491    1495    1504    1509    1524    1535    1543    1566    1570    1574    1578    1606
          1612    1618    1644    1655    1657    1670    1677    1684    1696    1705    1738    1745    1777    1787    1794
          1835    1840    1841    1902    1915    1987    1992    1993    2104    2109    2110    2172    2190    2214    2219
          2220    2315    2320    2321    2436    2441    2442    2463    2468    2469    2535    2550    2578    2583    2584
          2669    2674    2675    2694    2699    2700    2790    2802    2872    2877    2878    2960    2965    2966    3030
          3045    3116    3121    3122    3205    3210    3211    3360    3365    3366    3452    3463    3490    3495    3496
          3538    3543    3544    3608    3639    3730    3735    3736    3865    3870    3871    3945    3955    4026    4031
          4032    4092    4102    4162    4167    4168    4228    4241    4269    4274    4275    4332    4337    4338    4401
```

```
        4411    4471    4476    4477    4534    4560    4623    4628    4629    4716    4721    4722    4886    4891    4892
        5053    5058    5059    5124    5137    5200    5205    5206    5265    5270    5271    5410    5415    5416    5472
        5483    5550    5555    5556    5638    5658    5728    5733    5734    5760    5765    5766    5841    5846    5847
        5933    5957    6050    6055    6056    6124    6129    6130    6262    6267    6268    6340    6349    6420    6425
        6426    6495    6506    6593    6598    6599    6679    6689    6748    6753    6754    6845    6866    6929    6934
        6935    7013    7018    7019    7046    7066    7099    7104    7105    7176    7181    7182    7287    7292    7293
        7348    7364    7396    7401    7402    7473    7478    7479    7524    7529    7530    7598    7603    7604    7656
        7666    7671    7677    8265    8271    8278    8282    8286    8293    8297    8417    8423    8434    8440    8446
        8450    8456    8462    8469    8489    8760    8923    8931    8939    8947    8955    8963    8971    8979    8987
        8995    9003    9011    9019    9027    9035    9043    9051    9059    9067    9220    9221    9230    9231    9232
        9233    9234    9235    9236    9237    9238    9239
.PAGE    202     258     963    1181    1232    1263    1346    7700    7834    7862    7882    7919    8122    8237    8331
        8465    8477    8541    8850    9205    9242    9284
.RADIX  1840    1841    1992    1993    2109    2110    2219    2220    2320    2321    2441    2442    2468    2469    2583
        2584    2674    2675    2699    2700    2877    2878    2965    2966    3121    3122    3210    3211    3365    3366
        3495    3496    3543    3544    3735    3736    3870    3871    4031    4032    4167    4168    4274    4275    4337
        4338    4476    4477    4628    4629    4721    4722    4891    4892    5058    5059    5205    5206    5270    5271
        5415    5416    5555    5556    5733    5734    5765    5766    5846    5847    6055    6056    6129    6130    6267
        6268    6425    6426    6598    6599    6753    6754    6934    6935    7018    7019    7104    7105    7181    7182
        7292    7293    7401    7402    7478    7479    7529    7530    7603    7604
.REPT     12      16     148     240     246
.SBTTL    22      36     142     149     165     204     260    1184    1346    1466    1504    1524    1644    1696    1738
        1777    1902    2172    2535    2790    3030    3452    3608    3945    4092    4228    4401    4534    5124    5472
        5638    5933    6340    6495    6679    6845    7046    7348    7656    7702    7744    8480    8543    8611    8687
        8798    8852    8899    9129    9207    9222    9244
.TITLE    1
.WORD   148     215     218     219     220     221     224     225     226     227     228     229     238     240     241
         242     243     244     245     246     247     248     249     250     251    1286    1321    1322    1324    1328
        1329    1342    1343    1344    1345    7716    7719    8236    8678    8689    8690    8691    8846    8849    9096
        9101    9204    9274    9276   10609   10612   10615   10617   10619   10620   10623   10626   10628   10630   10632
       10634   10636   10638   10641
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DERPVB,DERPVB/CRF/SOL=DERPVB.P11/DS:ERFZ
RUN-TIME: 98 101 14 SECONDS
RUN-TIME RATIO: 488/214=2.2
CORE USED:  24K  (47 PAGES)
```