RP04/05/06

FUNCTION CONTROL PART 1

MD-11-DZRJI-A

EP DZRJI A DL A NOV 1976

COPYRIGHT 1976

FICHE 1 OF 2 MADE IN US

DZRJIA
SEQ

RP04/05/06

FUNCTION CONTROL PART 1

MD-11-DZRJI-A

EP-DZRJI-A-DL-A

COPYRIGHT © 1976

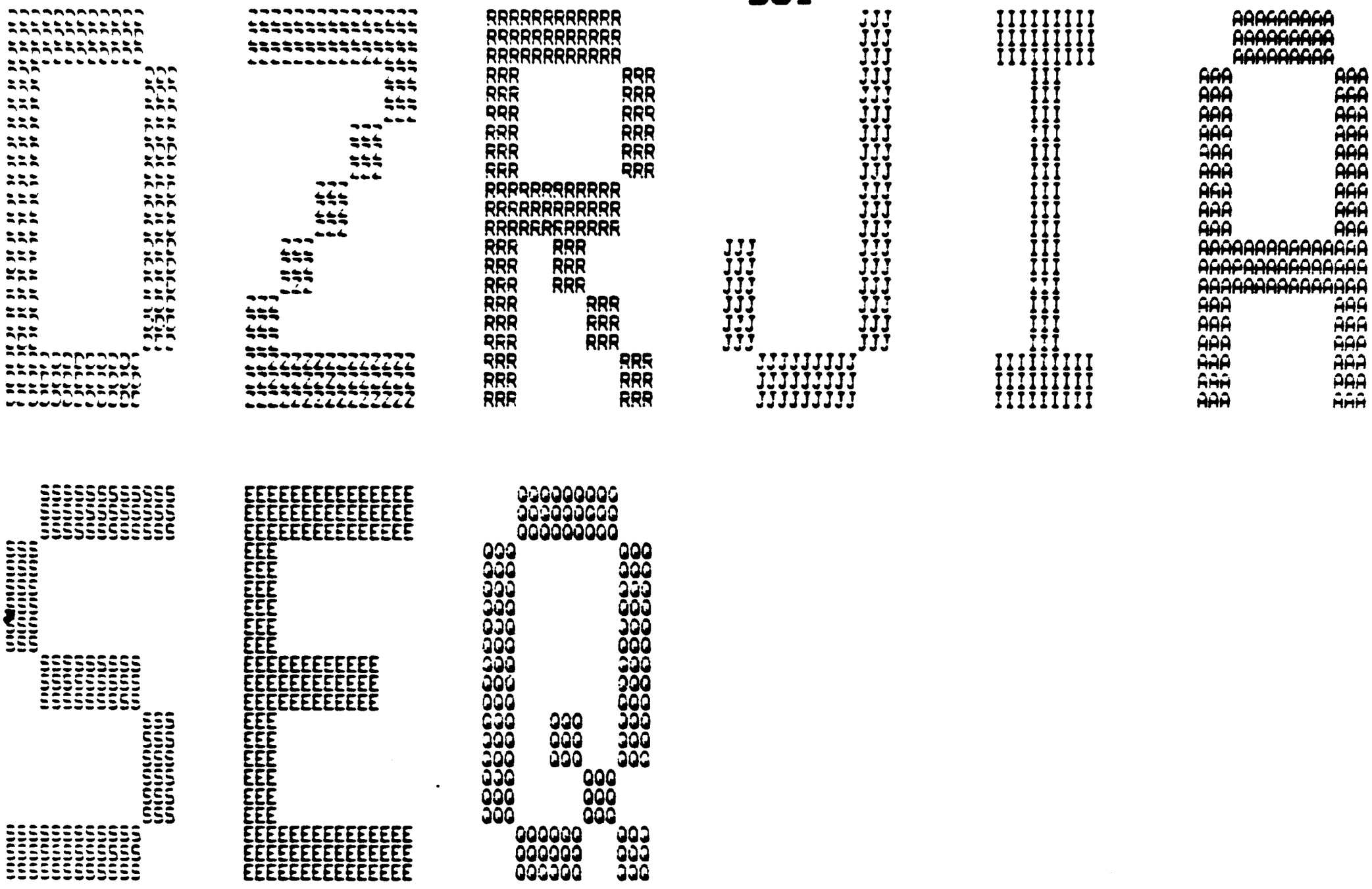
FICHE 2 OF 2

NOV 1976

digital

MADE IN USA

B01




```
_STEP_ .ersion 101(2111) Running on MTA140
 *START* User WELLINGHAM,A[400,3722]  Job  DZRJIA Seq. 5167 Date 07-May-76 14:24:43 Monitor     RV242A KI10 SYS#514 *START*
Request created: 07-May-76 14:17:27
File: PPSB:DZRJIA.SEQ<055>[400,3722] Created: 03-May-76 10:14:41 Printed: 07-May-76 14:24:45
QUEUE Switches:   PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:2000 /FORMS:NORMAL
```

IDENTIFICATION

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

PRODUCT NAME: RPC4/5/6 FUNCTIONAL CONTROLLER TEST PART I

DATE CREATED: MAY 1976

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: PETE BLACKSTONE

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITTMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITA EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORPORATION

D01

CONTENTS

SEG 0002

1. ABSTRACT
2. REQUIREMENTS
2:1 EQUIPMENT
2.2 STORAGE
2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
3.1 METHOD
4. STARTING PROCEDURE
4.1 CONTROL SWITCH SETTINGS
4.2 STARTING ADDRESS OR ADDRESSES
4.3 PROGRAM AND/OR OPERATOR ACTION
5. OPERATING PROCEDURE
5.1 OPERATIONAL SWITCH SETTINGS
5.2 SUB-ROUTINE ABSTRACTS
6. ERRORS
6.1 'FATAL' ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
8.1 EXECUTION TIME
8.2 STACK POINTER
8.3 OPERATOR SELECTABLE SCOPE LOOPS
8.4 PROGRAM REVISION HISTORY
9. PROGRAM DESCRIPTION

1.0 ABSTRACT

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

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04/5/6 DISK SYSTEM. THE RP04/5/6 DISK SYSTEM WILL CONSIST OF AN RH11 CONTROLLER, A DISK CONTROL LOGIC (DCL), A DEC 733 DISK DRIVE, AND ITS APPROPRIATE DISK PACK. THE DISK PACK NEED NOT BE FORMATTED. USED SECTION OF THE DISK SURFACE SHALL BE GOOD (HOLE FREE). THE SURFACE FOR THE FOLLOWING SECTORS MUST BE GOOD, THAT IS, FREE OF ANY HOLES OR SURFACE IRREGULARITY BEFORE ANY DATA ERROR CAN BE ATTRIBUTED TO THE LOGIC.

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

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY

2.3 PRELIMINARY PROGRAMS

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

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.2 STARTING PROCEDURE

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

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

START AT ADDRESS 200---FOR NORMAL RUN
START AT ADDRESS 210---FOR UNIT SELECTION
START AT ADDRESS 220---FOR NO MANUAL INTERVENTION

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

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

220 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE PROGRAM WILL NOT RUN THOSE TESTS THAT NEED MANUAL INTERVENTION. THIS IS RECOMMENDED ONLY FOR DEBUGGING WHERE THE ERROR IS NOT IN A TEST THAT REQUIRES MANUAL INTERVENTION

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.

2. SET STARTING ADDRESS ON THE SWITCH REGISTER

3. PRESS "LOAD ADDRESS".

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

5. PRESS "START".

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

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEIDNG RUN ON A SWITCHLES PROCESSOR (I. E.
AN 11/34) IT WILL DETERMINE THAT A HARWARE SWITCH REGISTER IS
NOT PRESENT, AND WILL USE "SOFTWARE" SWITCH REGISTER. THE
SETTINGS OF THE SWITCHES ARE CONTROLLED THROUGH A KEYBOARD
ROUTINE WHICH IS CALED BY TYPING A 'COBNTROL G'. THE PROGRAM
WILL RECOGNIZE A 'CONTROL G' AT ANY TIME EXCEPT WHEN IT IS AR
A HIGHER PRIORITY PROCESSING AN RP04/5/6 INTERRLPT. THE
"SOFTWARE" SWITCH VALUE S ARE ENTERED AS AN OCTAL NUMBER
IN RESPONSE TO PROMPTING FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

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

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

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

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

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

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

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

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

NOTE: SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR <7:0>
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7
HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE
ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH
0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPIATE
POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU
7 GIVE THE TEST NUMBER TO BE LOOPED ON. FOR EXAMPLE
WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP
ON TEST 10. HOWEVER THIS SETTING MUST BE DONE AT THE
BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10
WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER
AND OVER AGAIN. WHEN THIS SWITCH IS NOT SET THEN SWITCHES
0 THRU 7 HAVE THE MEAING ITS NAME INDICATES.
FOR EXAMPLE SWITCH 7 IS "STOP FURTHER COMPARES: THAT IS
IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A
DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.
FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN
ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW

WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS
OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT
ALL THE 256 WORDS. HOWEVER IF THIS WAS DONE WITH SWITCH 11
THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-
SEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY
THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER
EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH
IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: SEE SECTION 8.3

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

SWITCH 6 - TYPE ALL REGISTERS WITH ERROR IF SW08 IS LOW
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN
THIS SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS
INDICATED IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS
NOT SET AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO
AS THE NAME INDICATES. THAT IS ON FINDING AN ERROR INSTEAD
OF ONLY GIVING THE ERROR MESSAGE AND RELEVANT REGISTERS
AS WILL BE DONE IF SWITCH 11 IS NOT SET BUT WILL ALSO
GIVE ALL THE REGISTER CONTENTS (EXCEPT "DATA BUFFER" RHDB).

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES".

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER
PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE.
THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04/5/6
REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A
SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO
GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED
ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE
CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS

CANNOT BE CLEARED PRIOR TO THE START OF A TEST SEQUENCE - THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATIOR. IN ADDITION, THE TTY BELL WILL RING AND THE PROGRAM WILL HALT. IT IS SUGGESTED THAT IF THIS HAPPENS, THE OPERATOR LOAD ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT TO SOLVE THE PROBLEM. IF THE FILURE CONTINUES TO OCCUR, THERE ARE TWO OPTIONS OPEN TO THE OPERATOR:

1. LOOK IN THE TEST LISTING FOR THE 'HALT' INSTRUCTION AND REPLACE IT, PLUS THE TWO WORDS ("TYPE ,CPHALT") ABOVE WITH 'NOP'S. WITH TTY ERROR PRINTOUTS INHIBITED, A SCOPE LOOP CAN BE INITIATED FOR THE TEST IN QUESTION.

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

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

7.0 RESTRICTIONS

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

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

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

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS. FOR INSTRUCTIONS REGARDING THE USAGE OF THIS TECHNIQUE, HIT ↑C ANY TIME WHILE THE PROGRAM IS RUNNING. ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT THE PROGRAM GOES BACK TO CAN BE CHANGED.

THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
2. LOOP ON ERROR SWITCH MUST BE SET
3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT COMES TO THE END OF THE TEST UNDER CONSIDERATION.

AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN NORMAL OPERATION WILL CONTINUE.

8.4 PROGRAM REVISION HISTORY

9.0 PROGRAM DESCRIPTION

9.1 LOGIC DIVISION IN HARDWARE MODULES

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

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

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

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

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

9.2 DISK SURFACE USAGE

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

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

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

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

L01

SEQ 0010

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-DZRJI-A, RP04/5/6 FUNCT. CONT. TST-PT 1
****

COPYRIGHT 1976
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

TABLE OF CONTENTS
****

40 OPERATIONAL SWITCH SETTINGS
57 BASIC DEFINITIONS
168 TRAP CATCHER
178 ACT11 HOOKS
189 STARTING ADDRESSES
208 MEMORY MANAGEMENT DEFINITIONS
245 COMMON TAGS
303 ERROR POINTER TABLE
1322 REGISTER ADDRESSES
1505
1506 **DIAGNOSTIC CODE**
1507
1510 SETUP TESTS
1525 INITIALIZE THE COMMON TAGS
1612 GET VALUE FOR SOFTWARE SWITCH REGISTER
3019
3020 **DRIVE COMMAND TESTS**
3021
5056 READ/WRITE TESTS USING MEDIA
7454 SEEK TESTS
9770 WRITE CHECK DATA & WRITE PROTECT TESTS
10266

TABLE OF CONTENTS
****

10267 **SUBROUTINES**

10268

10271 END OF PASS ROUTINE

10312 JAM CURRENT CYLINDER ROUTINE

11114 SCOPE HANDLER ROUTINE

11177 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

11244 TYPE ROUTINE

11314 TTY INPUT ROUTINE

11542 READ AN OCTAL NUMBER FROM THE TTY

11595 ERROR HANDLER ROUTINE

11546 ERROR MESSAGE TYPEOUT ROUTINE

11977 BINARY TO OCTAL (ASCII) AND TYPE

11954 TRAP DECODER

11970 TRAP TABLE

11932 POWER DOWN AND UP ROUTINES

```
                3       COPYRIGHT (C) 1976
                        DIGITAL EQUIPMENT CORP.
                        MAYNARD, MASS. 01754

                        PROGRAM BY PETE BLACKSTONE

                        THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        PACKAGE (MAINDEC-11-DZQAC-CO),MAR 21, 1976.


                        DRIVE MUST BE LOCKED ON PORT A OR PORT B

                28      INTERNAL PROGRAM MACROS BEGIN HERE
                        ************************************

                33      *
                        *NOTE:  ALL MACRO CALLS BEGINNING WITH ".$" ARE SUPPLIED FROM AN
                                EXTERNAL SYSMAC.SML PACKAGE WHICH MUST BE MADE AVAILABLE
                                TO THE SOURCE PROGRAM AT ASSEMBLY TIME.


                        ********************************************************************
        4C              OPERATIONAL SWITCH SETTINGS
                        ********************************************************************

                41
                                SWITCH                  USE
                                ------                  ---------------------
                                  15                    HALT ON ERROR
                                  14                    LOOP ON TEST
                                  13                    INHIBIT ERROR TYPECUTS
                                  12                    RH70 CONTROLLER SELECT
                                  11                    INHIBIT ITERATIONS
                                  10                    BELL ON ERROR
                                   9                    LOOP ON ERROR
                                   8                    LOOP ON TEST IN SWR<7:0>

        53                         7                    STOP FURTHER COMPARES IF SW09 IS LOW
                                   6                    TYPE ALL REG. WITH ERROR IF SW8 LOW
                                   5                    MULT ADDR PLUG TEST IF SW09 IS LOW
```

```
                ;*****************************************************************
                ;BASIC DEFINITIONS
                ;*****************************************************************

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

                        MISCELLANEOUS DEFINITIONS

                        GENERAL PURPOSE REGISTER DEFINITIONS

                        PRIORITY LEVEL DEFINITIONS

                        "SWITCH REGISTER" SWITCH DEFINITIONS

                        DATA BIT DEFINITIONS (BIT00 TO BIT15)

                        BASIC "CPU" TRAP VECTOR ADDRESSES

                ;*****************************************************************
                ;TRAP CATCHER
                ;*****************************************************************

                        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

                ;*****************************************************************
                ;ACT11 HOOKS
                ;*****************************************************************

                ;*****************************************************************
                ;STARTING ADDRESSES
                ;*****************************************************************

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

                        STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE

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

```
                ;*********************************************************************
        208     ;MEMORY MANAGEMENT DEFINITIONS
                ;*********************************************************************

                210     KT11 VECTOR ADDRESS

                214     KT11 STATUS REGISTER ADDRESSES

                221     KERNEL "I" PAGE DESCRIPTOR REGISTERS

                232     KERNEL "I" PAGE ADDRESS REGISTERS

                ;*********************************************************************
        245     ;COMMON TAGS
                ;*********************************************************************

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

                ;*********************************************************************
        303     ;ERROR POINTER TABLE
                ;*********************************************************************

                305     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:

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

                323     ITEM1

                335     ITEM2

                347     ITEM3

                358     ITEM4

                371     ITEM5

                381     ITEM6

                391     ITEM7

                399     ITEM10
```

412 ITEM11
426 ITEM12
440 ITEM13
456 ITEM14
466 ITEM15
478 ITEM16
492 ITEM17
506 ITEM20
519 ITEM 21
526 ITEM 22
533 ITEM 23
543 ITEM 24
552 ITEM 25
563 ITEM 26
575 ITEM 27
586 ITEM 30
596 ITEM 31
607 ITEM 32
615 ITEM 33
625 ITEM 34
631 ITEM 35
641 ITEM 36
648 ITEM 37
659 ITEM 40
670 ITEM 41

```
                684     ITEM 42
                696     ITEM 43
                709     ITEM 44
                721     ITEM 45
                734     ITEM 46
                743     ITEM 47
                756     ITEM 50
                767     ITEM 51
                783     ITEM 52
                796     ITEM 53
                803     ITEM 54
                815     ITEM 55
                827     ITEM 56
                839     ITEM 57
                852     ITEM 60
                869     ITEM 61
                878     ITEM 62
                888     ITEM 63
                896     ITEM 64
                908     ITEM65
                920     ITEM66
                936     ITEM67
                947     ITEM70
                959     ITEM 71
                972     ITEM 74
```

```
                983     ITEM 73
                990     ITEM 74
                1001    ITEM 75
                1009    ITEM 76
                1021    ITEM 77
                1031    ITEM 100
                1040    ITEM 101
                1052    ITEM 102
                1064    ITEM    103
                1076    ITEM    104
                1090    RH11 REGISTERS
                1098    WORD COUNT REGISTER (RHWC)
                        EACH BIT IS CALLED BY BIT NUMBER
                1103    BUS ADDRESS REGISTER (RHBA)
                        EACH BIT IS CALLED BY BIT NUMBER
                1108    CONTROL AND STATUS REGISTER 2 (RHCS2)
                1127    DATA BUFFER REGISTER (RHDB)
                        EACH BIT IS CALLED BY BIT NUMBER
                1132    RP04 REGISTERS
                1137    CONTROL AND STATUS 1 REGISTER. (#00)
                1150    STATUS REGISTER (RHDS1) (#01)
                1169    ERROR REGISTER #01 (RHER1) (#02)
                1187    MAINTAINABILITY REGISTER (RHMR)(#03)
                1197    ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
                1208    DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
                        EACH BIT IS CALLED BY BIT NUMBER
                1215    DRIVE TYPE REGISTER (RHDT) (#06)
                        EACH BIT IS CALLED BY BIT NUMBER
```

```
                1222    LOOK-AHEAD REGISTER (RHLA) (#07)

                1241    ERROR REGISTER #2 (RHER2) (#10)

                1259    OFFSET REGISTER (RHOF) (#11)

                1273    DESIRED CYLINDER ADDRESS (RHCA) (#12)
                        EACH BIT IS CALLED BY BIT NUMBER.

                1279    CURRENT CYLINDER ADDRESS (RHCC) (#13)
                        EACH BIT IS CALLED BY BIT NUMBER

                1285    SERIAL NUMBER REGISTER (RHSN) (#14)
                        EACH IS CALLED BY BIT NUMBER

                1291    ERROR REGISTER #03 (RHER3) (#15)

                1305    ECC POSITION REGISTER (RHEC1) (#16)
                        EACH BIT IS CALLED BY BIT NUMBER

                1311    ECC PATTERN REGISTER (RHEC2) (#17)
                        EACH BIT IS CALLED BY BIT NUMBER

                        ;*****************************************************************
        1322    REGISTER ADDRESSES
                ;*****************************************************************

                1327    RP04/5/6 DISK I/O REGISTER LOCATED IN THE RH11 CONTROLLER

                1334    RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)

                1353    ADDITIONAL I/O REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

                1359    P-CLOCK (KW11-P) I/O REGISTERS

                1367    THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SNAPSHOTS
                        ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED

                1370    ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
                        FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND

                1375    THIS ASSUMES THAT A REGISTER SNAPSHOT HAS BEEN TAKEN WHICH IS NOT
                        ALWAYS THE CASE - IF QUESTIONABLE CONTENTS APPEAR IN THE REGISTER
                        PRINTOUTS, CHECK THE INLINE TEST CODE TO SEE IF THE REGISTER SNAPSHOT
                        REFLECTS THE CURRENT STATE OF THE MACHINE

                1407    FUNCTION EQUATES

                1409    TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
```

1432 DATA BUFFERS FOR READ/WRITE

1440 RESERVED CORE LOCATIONS

1461 TABLE FOR ATTENTION BITS
      ATTENTION TABLE

1470 FLAGS AND INTERNAL PROGRAM CONTROL WORDS

```
                ;*********************************************************************
1505            ;
                ;*********************************************************************

                ;*********************************************************************
1506            ;**DIAGNOSTIC CODE**
                ;*********************************************************************

                ;*********************************************************************
1507            ;
                ;*********************************************************************

                ;*********************************************************************
1510            ;SETUP TESTS
                ;*********************************************************************

                ;*********************************************************************
1525            ;INITIALIZE THE COMMON TAGS
                ;*********************************************************************

                ;*********************************************************************
1612            ;GET VALUE FOR SOFTWARE SWITCH REGISTER
                ;*********************************************************************

1627            ;IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
                ;IF SO MAKE 'WAT" TRAPS GO TO 'WAIT.P'
                ;IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
                ;IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
                ;IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'

1633            ;THE NEXT LINE IS TO BE ADDED LATER
                ;AND THE JUMP AND NOP REMOVED
                ;FOR NOW NO CLOCK WILL BE USED

1676            ;TEST 1  REFERENCE EACH REGISTER

1678            ;                REFERENCE EACH REGISTER BY A MOVE INSTRUCTION

1722            ;TEST 2  PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
```

```
                1724            CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES INTO IT

                1745    TEST 3  TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

                1747            THE NUMBER OF DRIVES PRESENT IS FOUND
                                BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
                                IN RHCS2 INCREMENTED FROM ZERO TO SEVEN

                1751            THE BITS SET IN RHAS SHOULD GIVE DRIVES PRESENT

                1753            THE DRIVE TYPE IS CHECKED TO BE AN RP04/RP06 AND THEN
                                UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'

                1813    SET UP UNITS TABLE

                1848    NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
                        OUT THE DEVICE TYPE

                1891    TEST 4  TYPE SERIAL NUMBER AND DRIVE TYPE

                1893            SET APPROPRIATE ATTTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
                                TYPE UNIT UNDER TEST

                1896            READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
                                TYPE THEM OUT AND PROCEED

                1899            TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
                                SWITCHES 0 THRU 7 AND RESTART

                1925    INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
                        & DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)

                2052    TEST 5  CHECK MOL TO BE HIGH

                2054            MAKE SURE THAT DRIVE IS ON LINE BEFORE STARTING PROGRAM
                                IF DRIVE IS OFFLINE, THEN AFTER TYPE OUT THE PROGRAM WILL
                                HANG FOR EVER WAITING FOR DRIVE TO GO ON LINE.

                2084    TEST 6  PROGRAM INTERRUPT

                2086            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

                2125    TEST 7  INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

                2127            PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
                                IE AND RDY IS SET. THIS SHOULD NOT ALLOW INTERRUPT
```

```
                2165    TEST 10 PACK ACKNOWLEDGE

                2167            IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED

                                IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                THEN THIS TEST IS NOT PERFORMED

                                IF NO ACT-11 MONITOR IS PRESENT
                                THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ON SUBSEQUENT PASSES THIS TEST IS NOT DONE

                                THIS TESTS THE ACKNOWLEDGE COMMAND=44
                                VV BIT - RHDS1 BIT #6
                                MOL BIT - RHDS1 BIT #12
                                DVA BIT - RHCS1 BIT #11
                                THE DRIVE IS STOPED MOL IS CHECKED TO BE 0
                                AND DVA SHOULD BE 0
                                THE DRIVE IS POWERED UP
                                VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1
                                GO SHOULD BE 0
                                ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED
                                PACK ACKNOWLEDGE IS ISSUED
                                ALL STORED REGISTERS SHOULD BE UNCHANGED
                                EXCEPT VV

                2202    THIS CODE CHECKS TO SEE IF MANUAL INTFRVENTION TESTS ARE OK

                2304    NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE

                2345    NOW COMPARE REGISTERS SO THAT NO REGISTERS
                        CHANGED EXCEPT VV IN RHDS1 AND IE IN RHCS1

                2378    TEST 11 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.

                2388    IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
                        OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN

                2409    NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE

                2444    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                2475    NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
                        WITH AFTER PACK ACKNOWLEDGE
```

```
                2517    TEST 12 MAKE CURRENT CYLINDER = 377

                2536    TEST 13 MAKE CURRENT CYLINDER = 777

                2553    TEST 14 ADDRESS PLUG CHANGE ERROR

                2555            CHECK PROPER ADDRESS PLUG FUNCTIONALITY
                                BY PULLING THE ADDRESS PLUG DURING A COMMAND ISSUED
                                IN DIAGNOSTIC MODE (TO GUARANTEE COMMAND IS STILL ACTIVE WHEN
                                PLUG IS PULLED) AND VERIFYING THAT THE DRIVE GOES OFF LINE

                2560            THE ADDRESS PLUG IS THEN REPLACED AND RETURN OF THE PROPER
                                RESPONDING DRIVE IS VERIFIED, AS WELL AS THE FACT THAT
                                ATTENTION BITS COME UP IN THE PROPER BIT LOCATION(?????), AND THE
                                ADDRESS CHANGE ERROR (ACE) - RHER3 BIT #13 IS SET, AS WELL AS
                                SC,TRE,ERR,AND MCPE; VV IS RESET AND RHCC = 000 (DRIVE RECALIBRATED)

                2566            IN ADDITION VERIFICATION IS ALSO MADE THAT NO OTHER DRIVE
                                NUMBERS APPEAR ON THE BUSS AFTER THE PLUG IS REPLACED. (??????)

                2597    THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK

                2611    SET DIAGNOSTIC MODE TO ENABLE A COMMAND ACTIVE WHILE
                        THE PLUG IS PULLED

                2616    TAKE AN INITIAL REGISTER SNAPSHOT

                2626    ISSUE A COMMAND AND THE 'GO' BIT (NOT POSITIONING COMMAND)
                        TO VERIFY COMMAND ABORT IF PLUG IS PULLED

                2631    ISSUE SOME CLOCKS TO GET THE COMMAND STARTED
                        (USE "SEARCH" WITH "DTETST" FLAG UP TO STOP CLOCKING ?)

                2648    CHECK THAT THE UNIT NO. UNDER TEST HAS GONE OFFLINE

                2659    NOW REPLACE THE ADDRESS PLUG

                2674    CHECK THAT THE ORIGINAL UNIT HAS COME BACK ON LINE

                2689    AFTER THE PLUG CHANGE

                2733    TAKE A NEW REGISTER SNAPSHOT AND
                        COMPARE THE REGISTER CONTENTS WITH EXPECTED VALUES

                2751    NOW CLEAR OUT THE CONTROLLER AND DRIVE

                2760    CHANGE THE REGISTER SNAPSHOT TO EXPECTED VALUES AFTER THE CLEAR

                2807    TAKE ANOTHER REGISTER SNAPSHOT AND COMPARE RESULTS
                        WITH THE EXPECTED VALUES
```

```
                2824    ;USE NED METHOD TO VERIFY
                        ;THAT ATTENTION BIT COMES UP IN THE PROPER LOCATION ??)

                2830    SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
                        CAUSED BY PULLING THE PLUG

                2855    TEST 15 CHECK ALL ADDRESS PLUG ADDRESSES

                2857            CHECK THAT ALL ADDRESS PLUG NUMBERS FUNCTION PROPERLY

                2859            THIS TEST IS DONE FOR MEMOREX RP04'S AND RP05'S ONLY
                                THIS TEST IS SELECTED BY SW#5 AND THE DEFAULT IS NOT TO DO IT

                2877    CHECK TO SEE IF THIS TEST HAS BEEN SELECTED WITH SW5

                2895    THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE Ok

                2927    CHANGE ADDRESS PLUG ON THE UNIT UNDER TEST

                2946    HOUSEKEEPING

                2958    ATTEMPT TO ADDRESS THE NEW UNIT NUMBER

                2979    CHECK IF ALL UNIT NUMBERS HAVE BEEN TRIED

                2989    SET 'VV'IN RHDS1 AFTER RESET FROM THE RECALIBRATE
                        CAUSED BY PULLING THE ADDRESS PLUGS OUT

                ;*******************************************************************
        3019
                ;*******************************************************************

                ;*******************************************************************
        3020    ;**DRIVE COMMAND TESTS**
                ;*******************************************************************

                ;*******************************************************************
        3021
                ;*******************************************************************

                3029    TEST 16 NO OPERATION FUNCTION TEST
                                ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NOP"=0
                                IS GIVEN NO CHANGE SHOULD HAPPEN
                                ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
                                IS GIVEN NO CHANGE SHOULD HAPPEN

                3054    NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION

                3087    NOW COMPARE REGISTERS BEFORE NO-OP COMMAND
                        WITH AFTER NO-OP COMMAND
```

```
                3107    NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES

                3135    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                3169    NOW SAVE REGISTERS FOR COMPARISON AFTER A NO-OP

                3202    CHANGE REGISTERS TO EXPECTED VALUES

                3223    NOW COMPARE REGISTERS BEFORE NO-OP WITH
                        AFTER NO-OP COMMAND

                3246    TEST 17 DRIVE CLEAR
                                ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                THEN A DRIVE CLEAR IS GIVEN
                                THEN ALL REGISTERS ARE CHECKED TO HAVE APPROPIAVE VALUE

                3266    WRITE ALL WRITABLE REGISTER BITS

                3288    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                3338    NOW LOAD 'SAVERE' REGISTER SNAPSHOT WITH EXPECTED VALUES

                3436    NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
                        EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
                        WITH SAVERE TABLE

                3461    TEST 20 READ-IN-PRESET
                                IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED

                        IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                        THEN THIS TEST IS NOT PERFORMED

                        IF NO ACT-11 MONITOR IS PRESENT THEN
                        THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                        ON SUBSEQUENT PASSES THIS TEST IS NOT DONE


                        THIS TESTS THE READ-IN-PRESET COMMAND
                        FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO
                        RESET VV-BIT #6 IN RHDS1
                        THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                        WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                        ATA FOR DRIVE UNDER TEST IS MADE = 0


                        THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
                        THEN ALL REGISTERS ARE TESTED
                        THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
                        RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
                        FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
                        ALL OTHER REGISTERS SHOULD BE UNCHANGED
```

```
                3497    THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK

                3544    NOW INITIALIZE ALL THE REGISTERS

                3565    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                3595    NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND

                3643    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                3670    NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
                        THE EXPECTED VALUE AFTER A READ-IN COMMAND
                        COMPARISONS ARE MADE

                3702    TEST 21 READ-IN-PRESET

                3704            THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT
                                THAT IT DOES NOT TEST THE SETTING OF VV

                                THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT
                                THEN THE PREVIOUS TEST WILL NOT BE PERFORMED
                                THIS TESTS THE READ-IN-PRESET COMMAND
                                ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                ATA FOR DRIVE UNDER TEST IS MADE = 0

                                THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
                                THEN ALL REGISTERS ARE TESTED
                                THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
                                RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
                                FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
                                ALL OTHER REGISTERS SHOULD BE UNCHANGED

                3737    INITIALIZE ALL THE REGISTERS

                3767    NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND

                3815    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                3842    NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
                        THE EXPECTED VALUE AFTER A READ-IN COMMAND
                        COMPARISONS ARE MADE

                3883    TEST 22 MAKE CURRENT CYLINDER = 777

                3902    TEST 23 MAKE CURRENT CYLINDER = 377

                3921    TEST 24 RECALIBRATE COMMAND
```

```
                3923            ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
                                THEN A RECALIBRATE = 6 COMMAND IS GIVEN

                3926            NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0

                3962    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                3991    NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE

                4022    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                4054    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

                4074    NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND

                4105    TEST 25 MAKE CURRENT CYLINDER = 777

                4124    TEST 26 MAKE CURRENT CYLINDER = 377

                4142    TEST 27 RECALIBRATE COMMAND

                4144            ALL POSSIBLE REGISTERS ARE FILLED WITH 0
                                THEN A RECALIBRATE =6 COMMAND IS GIVEN

                4147            NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0

                4191    NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE

                4222    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                4245    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

                4269    NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND

                4288    TEST 30 UNLOAD COMMAND

                4290            IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED

                                IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                THEN THIS TEST IS NOT PERFORMED

                                IF NO ACT-11 MONITOR IS PRESENT
                                THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ON SUBSEQUENT PASSES THIS TEST IS NOT DONE


                                ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
                                THEN AN UNLOAD COMMAND =2 IS GIVEN
                                NO REGISTERS SHOULD CHANGE EXCEPT MOL SHOULD=0
                                THEN THE DRIVE IS POWERED UP BY OPERATOR
                                AND A PACK ACKNOWLEDGE COMMAND (ALREADY TESTED)
                                SETS VV-IN RHDS1
```

```
                4311    THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK

                4337    THIS SETTING OF VV IS FOR LOOP ON ERROR ONLY
                        WHERE UNLOAD TAKES EFFECT AND CYCLE UP BRINGS VV DOWN

                4364    SET VV IN RHDS1 WITH PACK ACKNOWLEDGE

                4392    LOAD ALL POSSIBLE REGISTERS WITH ONES

                4434    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                4465    NOW SAVE REGISTERS FOR COMPARISON AFTER UNLOAD

                4499    COMPARE CONTENTS OF RHCS1 AND RHDS1, WHICH WERE SAVED
                        DURING THE UNLOAD COMMAND, WITH THE EXPECTED RESULTS

                4547    NOW CHANGE REGISTERS SAVED BEFORE UNLOAD COMMAND
                        TO EXPECTED VALUES AFTER UNLOAD COMMAND

                4550     - AGAIN 'MOL' $ 'VV' ARE DON'T CARES

                4585    NOW COMPARE REGISTERS AFTER THE UNLOAD COMMAND
                        WITH EXPECTED VALUES

                4624    SET VV IN RHDS1 AFTER RESET FROM ACTUATING
                        THE STANDBY SWITCH AND CYCLING UP (MOL = 1)

                4646    TEST 31 OFFSET AND RETURN TO CENTER LINE COMMAND

                4648            THIS TESTS TWO COMMANDS: (1)OFFSET, (2)RETURN-TO-CENTER-LINE

                4650            ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF)
                                AND AN OFFSET IS GIVEN
                                ALL REGISTERS ARE COMPARED, ONLY ATA SHOULD SET
                                ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED

                4655            THEN A RETURN-TO-CENTER-LINE IS GIVEN
                                ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
                                AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)
                                ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED

                4660            THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER
                                VALUES OF 1 TO 377 IE. 377 TIMES

                4713    THE OFFSET REGISTER WILL BE INCREMENTED FROM 0 TO 377

                4721    NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

                4748    NOW SAVE REGISTERS FOR COMPARISON AFTER OFFSET
```

```
                4792    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                4814    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

                4835    NOW COMPARE REGISTERS AFTER AN OFSET COMMAND

                4856    NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN

                4869    NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND

                4902    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                4934    NOW CHANGE SAVED REGISTER TO EXPECTED VALUE

                4958    NOW COMPARE REGISTERS AFTER RETURN-TO-CENTER-LINE

                4991    TEST 32 OFFSET COMMAND

                4993            THIS TEST WILL ONLY GIVE REPEATED OFFSETS

                ;*****************************************************************
        5056    ;READ/WRITE TESTS USING MEDIA
                ;*****************************************************************

                5058    TEST 33 WRITE/READ HEADER AND DATA (0'S)

                5060            WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                                TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS 256 WORDS
                                OF 0
                                THEN READ HEADER AND DATA FOR ABOVE.
                                WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                10000,0,0,0, AND 256 OF 0
                                THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
                                THE GO BIT, AND ALL THE REGISTERS ARE SAVED
                                THEN GO IS GIVEN FOR WRITE HEADER AND DATA

                                THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGED
                                THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED


                                NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
                                WITH ALL ONES, COMMAND IS LOADED INTO REGISTERS EXCEPT
                                GO BIT AND ALL REGISTERS ARE SAVED
                                GO IS GIVEN FOR THE READ COMMAND

                                ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                THEN THE READ DATA IS COMPARED.
```

5092 FILL WRITE FROM BUFFER WITH HEADER

5103 FILL WRITE FROM BUFFER WITH DATA

5111 NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
CHANGE WRITE FROM BUFFER

5130 NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

5150 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

5182 ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC

5192 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

5216 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

5228 NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
WITH REGISTERS AFTER COMMAND

5248 NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
NOTHING GOT CHANGED

5264 NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
READ INTO BUFFER IS FILLED WITH ONES

5279 NOW FILL COMMAND

5299 NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA

5339 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

5363 CHANGE SAVED REGISTERS TO EXPECTED VALUES

5375 COMPARE REGISTER BEFORE READ HEADER AND DATA
WITH REGISTERS AFTER COMMAND

5394 NOW READ INTO BUFFER WILL BE CHECKED TO SEE
THE READ WAS GOOD

5415 TEST 34 READ DATA (0'S)

5417 THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST
THE WRITE FROM BUFFER IS FILLED WITH ALL 0S
THE COMMAND IS FILLED, THEN ALL REGISTERS SAVED
THEN READ DATA COMMAND IS GIVEN
ALL REGISTERS ARE COMPARED FOR PROPER VALUES
READ DATA INTO 'READ INTO' BUFFER IS CHECKED

5434 FILL WRITE FROM BUFFER WITH EXPECTED DATA

5442 NOW THE READ DATA COMMAND WILL BE FILLED

5461 NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

5501 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

5525 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

5537 NOW COMPARE REGISTERS BEFORE READ DATA WITH
AFTER COMMAND

5553 NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
WAS GOOD

5572 TEST 35 WRITE/READ DATA (1'S & 125252)

5574 THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0
TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES
THIS SECTOR IS FORMATED BY PREVIOUS TEST
THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
SAME CYLINDER, TRACK, SECTOR, KEYS
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
200 ONES AND 56 125252
THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS
AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES
AND 56 OF 377.
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
GO IS GIVEN

ALL REGISTERS ARE CHECKED
READ DATA IS CHECKED

5608 NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252

5621 NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

```
                5636    NOW WRITE DATA COMMAND WILL BE LOADED

                5655    NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA

                5687    ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC

                5697    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                5721    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                5733    NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
                        AFTER COMMAND

                5752    NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

                5767    NOW A READ DATA COMMAND WILL BE GIVEN

                5769    FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF 377

                5788    FILL WRITE FROM BUFFER WITH 200 ONES AND 56 OF 377

                5801    NOW FILL COMMAND

                5820    NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

                5860    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                5884    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                5896    COMPARE REGISTERS BEFORE READ DATA COMMAND
                        WITH REGISTERS AFTER COMMAND

                5914    NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

                5934    TEST 36 WRITE/READ DATA (125252)

                5936            THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252
                                THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                SAME CYLINDER, TRACK, SECTOR, KEYS
                                WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                256 OF 125252
                                THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATI


                                THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

                                THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS
                                WRITE FROM BUFFER IS FILLED WITH
```

256 OF 125252
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
GO IS GIVEN

ALL REGISTERS ARE CHECKED
READ DATA IS CHECKED

5970 NOW FILL WRITE FROM BUFFER - 256 OF 125252

5978 NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

5988 NOW WRITE DATA COMMAND WILL BE LOADED

6007 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA

6039 ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC

6049 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

6073 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

6085 NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
AFTER COMMAND

6104 NOW WRITE FROM BUFFER WILL CHECKED FOR NO CHANGE

6119 NOW A READ DATA COMMAND WILL BE GIVEN
FILL READ INTO BUFFER WITH 256 ZEROS

6134 FILL WRITE FROM BUFFER WITH 256 OF 125252

6142 NOW FILL COMMAND

6161 NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

6201 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

6225 CHANGE SAVED REGISTERS TO EXPECTED VALUES

6237 COMPARE REGISTERS BEFORE READ DATA COMMAND
WITH REGISTERS AFTER COMMAND

6255 NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

```
                6273    TEST 37 WRITE/READ DATA (052525)

                6275            THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                                THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                SAME CYLINDER, TRACK, SECTOR, KEYS
                                WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                200 OF 052525 AND 56 OF 377
                                THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATI


                                THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

                                THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                                AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                                WORDS OF 52525 AND 56 WORDS OF 0
                                THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                GO IS GIVEN


                                ALL REGISTER ARE CHECKED
                                READ DATA IS CHECKED

                6309    NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377

                6322    NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
                        WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                        CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

                6337    NOW WRITE DATA COMMAND WILL BE LOADED

                6356    NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA

                6388    ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC

                6398    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                6422    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                6434    NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
                        AFTER COMMAND

                6453    NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

                6468    NOW A READ DATA COMMAND WILL BE GIVEN
```

```
                6470    FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES

                6489    FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0

                6502    NOW FILL COMMAND

                6521    NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

                6561    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                6585    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                6597    COMPARE REGISTERS BEFORE READ DATA COMMAND
                        WITH REGISTERS AFTER COMMAND

                6615    NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

                6636    TEST 40 WRITE/READ DATA USING UNIBUS B

                6638            THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
                                IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
                                THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                                THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                SAME CYLINDER, TRACK, SECTOR, KEYS
                                THESE COMMANDS USE UNIBUS B FOR DATA
                                WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                200 OF 052525 AND 56 OF 377
                                THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA


                                THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

                                THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                                AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                                WORDS OF 52525 AND 56 WORDS OF 0
                                THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                GO IS GIVEN


                                ALL REGISTER ARE CHECKED
                                READ DATA IS CHECKED

                6676    CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
```

```
6685    NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377

6698    NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
        WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
        CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

6713    NOW WRITE DATA COMMAND WILL BE LOADED

6734    NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA

6765    ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC

6775    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
        R0 AND R5 IMMEDIATELY AFTER GO

6800    CHECK IF NEM NON EXISTANT MEMORY IS SET
        IF SET IT MEANS UNIBUS B IS NOT CONNECTED
        SO THIS TEST IS NOT PERFORMED

6822    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

6834    NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
        AFTER COMMAND

6854    NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

6870    NOW A READ DATA COMMAND WILL BE GIVEN
        FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES

6890    FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0

6903    NOW FILL COMMAND

6924    NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

6963    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
        R0 AND R5 IMMEDIATELY AFTER GO

6987    CHANGE SAVED REGISTERS TO EXPECTED VALUES

6999    COMPARE REGISTERS BEFORE READ DATA COMMAND
        WITH REGISTERS AFTER COMMAND

7018    NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

7043    TEST 41 IMPLIED SEARCH

7045            ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEARCH
                A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTOR
                OF WORDS
                WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS
                266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1
                THEN READ HEADER AND DATA FOR ABOVE SECTOR 1 ONLY
```

WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
10000,0,0,0 AND 256 OF 0, 10000,1,0,0, AND 2 OF 1
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 CHANGE
THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED

NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000,1,0,0,1,1
AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT
GO AND IE THEN ALL REGISTERS ARE SAVED
GO IS GIVEN FOR THE READ COMMAND, 256 WORDS

ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
THEN THE READ DATA IS COMPARED

7081 FILL WRITE FROM BUFFER WITH HEADER

7091 FILL WRITE FROM BUFFER WITH DATA

7098 FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER

7108 FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA

7115 NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
CHANGE WRITE FROM BUFFER.

7147 NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

7167 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

7198 ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC

7208 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

7232 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

7244 NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
WITH REGISTERS AFTER COMMAND

7264 NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
NOTHING GOT CHANGED

```
                7280    NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
                        FOR SECTOR 1, 256 WORDS
                        READ INTO BUFFER IS FILLED WITH ONES

                7295    WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA

                7313    NOW FILL COMMAND

                7333    NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA

                7365    ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR 760 MICRO SECONDS

                7375    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                7399    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                7411    COMPARE REGISTERS BEFORE READ HEADER AND DATA
                        WITH REGISTERS AFTER COMMAND

                7431    NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                        THAT READ WAS GOOD

                7452    TEST 42 IMPLIED SEEK TO CYL 001

                ;*******************************************************************
        7454    ;SEEK TESTS
                ;*******************************************************************

                7456            ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM CYLINDER 0
                                TO CYLINDER 1. A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN
                                ONE SECTOR OF WORDS


                                WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                                TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS
                                (4 HEADER, 256 DATA=1125, 4 HEADER 2 DATA = 2000
                                THEN READ HEADER AND DATA FOR ABOVE, TRACK 0, SECTOR 0, CYL=1
                                WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                10000,0,0,0 AND 256 OF 1125, 10001,0,0,0, AND 2 OF 2000
                                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 CHANGE
                                THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED


                                NOW FOR READ COMMAND READ INTO BUFFER IS FILLED
                                WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH
                                10001,0,0,0 AND 2 OF 2000,254 OF ZEROS COMMAND IS LOADED
                                INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED
                                GO IS GIVEN FOR THE READ COMMAND
```

ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
THEN THE READ DATA IS COMPARED.

7494 FILL WRITE FROM BUFFER WITH HEADER

7504 FILL WRITE FROM BUFFER WITH DATA

7511 FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER

7521 FILL WRITE FROM BUFFER WITH NEXT TRACK DATA

7529 NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
CHANGE WRITE FROM BUFFER.

7560 NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

7580 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA1

7612 ONE REVOLUTION = 16670 MICRO1 SEC, ONE SECTOR = 760 MICRO1 SEC

7622 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

7646 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

7664 NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
WITH REGISTERS AFTER COMMAND

7684 NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
NOTHING GOT CHANGED

7700 NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
READ INTO BUFFER IS FILLED WITH ONES

7715 WRITE FROM BUFFER IS FILLED WITH 10001,0,0,0,2000,2000, AND 254 OF 0

7732 NOW FILL COMMAND

7751 NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA

7791 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO

7815 CHANGE SAVED REGISTERS TO EXPECTED VALUES

7826 COMPARE REGISTERS BEFORE READ HEADER AND DATA
WITH REGISTERS AFTER COMMAND

```
                7845    NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                        THAT READ WAS GOOD

                7867    TEST 43 SEEK & WRT TEST (CYL = 0-10)

                7869            THIS TEST GETS THE HEADS OUT TO CYLINDER 10 NOT BY ONE
                                SEEK BUT BY TEN IMPLIED SEEKS ONE CYLINDER AT A TIME

                                THIS TEST STARTS WITH A (ALREADY TESTED) RECALIBRATE
                                THAT IS CYLINDER ZERO.  THEN ON CYLINDER 0 SECTOR
                                #21 TRACK #18 IT WRITES 266 WORDS THERE BY GETTING
                                THE HEAD TO CYLINDER 1 THEN IT WRITES 266 WORDS ON
                                CYLINDER 1 SECTOR #21 TRACK #18 THERE BY GETTING
                                THE HEADS TO CYLINDER 2
                                THIS IS REPEATED 10 TIMES GETTING THE
                                HEADS TO CYLINDER 10
                                THEN A SEEK COMMAND IS GIVEN TO CYLINDER 0
                                AND DATA ALREADY WRITTEN IS CHECKED

                7893    THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM
                        CYLINDER 0
                        THESE LOCATIONS ARE CHANGED DURING TEST TO ENABLE
                        GOING TO NEXT CYLINDER

                7909    THIS IS TO GET THE HEADS TO CYLINDER 0

                7940    FILL WRITE FROM BUFFER WITH HEADER

                7951    FILL WRITE FROM BUFFER WITH DATA

                7959    FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER

                7970    FILL WRITE FROM BUFFER WITH NEXT TRACK DATA

                7978    THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

                7998    SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

                8030    ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760 MICRO SEC.
                        MAX TIME ALLOWED = ONE REVOLUTION + SEEK + 2 SECTORS
                        MIN TIME ALLOWED = 2 SECTORS + SEEK

                8042    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                8056    NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES

                8085    COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
                        WITH REGISTERS AFTER COMMAND
```

```
                9103    SETUP TO READ HEADER AND DATA FOR NEXT TRACK
                        FILL READ INTO BUFFER WITH ALL ONES

                9118    FILL WRITE FROM BUFFER WITH EXPECTED DATA

                9141    FILL COMMAND INTO REGISTERS

                9161    SAVE REGISTERS FOR COMPARISON AFTER READ HEADER
                        AND DATA

                9202    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                9226    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                9238    COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
                        REGISTERS AFTER COMMAND

                9257    READ INTO BUFFER IS CHECKED FOR PROPER READ

                8276    THE HEADS HAVE ADVANCED ONE CYLINDER BY AN IMPLIED
                        SEEK
                        CHANGES WILL BE MADE TO ENABLE GOING TO THE NEXT
                        CYLINDER AND THEN THE ABOVE WILL BE REPEATED
                        TILL CYLINDER 10 IS REACHED

                8297    THE HEADS ARE NOW AT CYLINDER 10
                        ALL REGISTERS WILL BE SAVED AND A SEEK WILL BE GIVEN
                        TO CYLINDER 0
                        FILL REGISTERS FOR A SEEK COMMAND

                8312    SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND

                8337    SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SEC

                8345    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                8370    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                8394    COMPARE REGISTERS AFTER A SEEK COMMAND

                8411    AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
                        NO ERROR BITS
                        A READ HEADER AND DATA WILL BE DONE ON CYLINDER 0
                        SECTOR 21 TRACK 18, EXPECTED DATA IS 1125
                        FOR 10 WORDS
                        CLEAR READ INTO BUFFER WITH ALL ONES
```

```
                8425    FILL WRITE FROM BUFFER WITH EXPECTED DATA

                8445    FILL READ HEADER AND DATA COMMAND FOR 10 WORDS

                8493    CHECK READ WORDS

                8513    TEST 44 SEEK & READ TEST (CYL = 009)

                8515            THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE
                                PREVIOUS TEST.  AT THIS POINT THE HEADS ARE ON
                                CYLINDER 0

                                ALL REGISTERS ARE SAVED
                                A SEEK IS GIVEN TO CYLINDER 9
                                ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                DATA WRITTEN ON CYLINDER 9 IS CHECKED

                8533    THIS GETS HEADS TO CYLINDER 0

                8559    FILL REGISTERS FOR A SEEK COMMAND

                8564    SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND

                8589    SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC

                8598    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                8622    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                8646    COMPARE REGISTERS AFTER A SEEK COMMAND

                8653    AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
                        NO ERROR BITS
                        A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
                        SECTOR 21 TRACK 18, EXPECTED DATA IS 23125
                        FOR 20 WORDS
                        CLEAR READ INTO BUFFER WITH ALL ONES

                8682    FILL    WRITE FROM BUFFER WITH EXPECTED DATA

                8703    FILL READ HEADER AND DATA COMMAND FOR 10 WORDS

                8751    CHECK READ WORDS
```

```
                ;*********************************************************************
        8770    ;WRITE CHECK DATA & WRITE PROTECT TESTS
                ;*********************************************************************

                8773    TEST 45 WRITE CHECK HEADER AND DATA

                8775            WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PER WORD
                                TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266
                                CONSISTING OF
                                10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                                10 WORDS OF 177777
                                10 WORDS OF 0
                                10 WORDS OF 052525
                                10 WORDS OF 125252
                                16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)
                                16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                                174 WORDS OF 377
                                4 WORDS OF HEADER
                                2 WORDS OF 12345


                                FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND
                                DATA COMMAND
                                CHECK FOR NO ERRORS
                                THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA
                                CHECK FOR NO ERRORS
                                THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN

        8811    GET HEADS TO CYLINDER 5

        8846    FILL WRITE FROM BUFFER WITH HEADER

        8856    10 WORDS OF OF THE FOLLOWING DATA
                 12344,17777,0,52525,125252

        8885    FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>

        8897    FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>

        8907    FILL REST OF DATA

        8929    READ INTO BUFFER WILL BE CLEARED

        8936    THE WRITE HEADER AND DATA COMMAND WILL BE LOADED

        8955    NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

        8993    NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
```

```
                9004    NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
                        WITH REGISTERS AFTER COMMAND

                9024    NOW FILL COMMAND FOR READ

                9050    NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA

                9089    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                9100    COMPARE REGISTERS BEFORE READ HEADER AND DATA
                        WITH REGISTERS AFTER COMMAND

                3121    NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                        THAT READ WAS GOOD

                9140    A WRITE,READ HAS BEEN SUCCESSFULLY COMPLETED
                        NOW A WRITE CHECK HEADER AND DATA WILL BE GIVEN
                        FILL THE WRITE CHECK HEADER AND DATA

                9168    SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK

                9209    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                9233    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                9245    COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
                        WITH REGISTERS AFTER COMMAND

                9269    TEST 46 WRITE CHECK DATA
                        THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
                        WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
                        TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
                        CONSISTING OF
                        10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                        10 WORDS OF 177777
                        10 WORDS OF 0
                        10 WORDS OF 052525
                        10 WORD OF 125252
                        16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
                        16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                        174 WORDS OF 377
                        2 WORDS OF 12345


                        FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
                        THEN THE ABOVE WRITE CHECK DATA IS GIVEN

                9302    GET HEADS TO CYLINDER 5
```

```
9337    10 WORDS OF EACH 12344,17777,0,52525,125252

9365    FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>

9377    FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>

9386    FILL REST OF DATA

9399    FILL THE WRITE CHECK HEADER AND DATA

9417    SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK

9456    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
        R0 AND R5 IMMEDIATELY AFTER GO

9480    CHANGE SAVED REGISTERS TO EXPECTED VALUES

9492    COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
        WITH REGISTER AFTER COMMAND

9516    TEST 47 WRITE CHECK DATA USING UNIBUS B
                THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
                IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
                THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
                WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
                TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
                CONSISTING OF
                10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                10 WORDS OF 177777
                10 WORDS OF 0
                10 WORDS OF 052525
                10 WORD OF 125252
                16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
                16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                174 WORDS OF 377
                2 WORDS OF 12345


                FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
                THEN THE ABOVE WRITE CHECK DATA IS GIVEN

9546    CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

9562    GET HEADS TO CYLINDER 5

9598    10 WORDS OF EACH 12344,17777,0,52525,125252

9626    FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
```

```
                9639    FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>

                9647    FILL REST OF DATA

                9659    FILL THE WRITE CHECK HEADER AND DATA

                9681    SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK

                9702    SET PORT SELECT

                9721    COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                        R0 AND R5 IMMEDIATELY AFTER GO

                9745    CHANGE SAVED REGISTERS TO EXPECTED VALUES

                9757    COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
                        WITH REGISTER AFTER COMMAND

                9785    TEST 50 WRITE PROTECT OPERATION
                                IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED

                                IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                THEN THIS TEST IS NOT PERFORMED

                                IF NO ACT-11 MONITOR IS PRESENT
                                THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ON SUBSEQUENT PASSES THIS TEST IS NOT DONE

                                WRITE FROM BUFFER IS FILLED WITH ALL ONES AND
                                SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH
                                ALL ONES
                                ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS
                                PRESSED AND ALL REGISTERS ARE CHECKED.
                                WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS
                                ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0  70. WORDS
                                ALL REGISTERS ARE CHECKED
                                THE SAME SECTOR IS READ AND DATA COMPARED TO SEE
                                THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL
                                ONES AND NOT 377)
                                THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK
                                WRITE LOCKS AND ALL REGISTERS ARE COMPARED

                9813    THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK

                9834    FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES
                        FILL WRITE FROM BUFFER

                9842    FILL WRITE DATA COMMAND
```

```
                9881    TIME IS NOT CRITICAL

                9890    SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT

                9891    BUTTON HAS BEEN HIT

                9911    THE ONLY REGISTER THAT SHOULD  CHANGE IS RHDS1 - BIT #11
                        -WRL

                9921    COMPARE ALL REGISTERS BEFORE WRITE WAS LOCKED
                        OUT WITH REGISTER VALUES AFTER WRITE WAS LOCKED OUT

                9941    NOW A WRITE WILL BE ATTEMPTED WITH WRITE LOCKED
                        OUT BY BUTTON
                        FILL WRITE FROM BUFFER WITH 377

                9958    TRY A ONE WORD WRITE

                9976    SAVE REGISTERS

                9998    TIME IS NOT CRITICAL

                10007   CHANGE SAVED REGISTERS TO EXPECTED VALUE

                10048   COMPARE REGISTERS BEFORE WRITE WAS ATTEMPTED
                        WITH REGISTERS AFTER ATTEMPT

                10069   NOW A READ WILL BE DONE TO DETERMIN THAT
                        READS CAN BE DONE WITH WRITE LOCKED OUT AND
                        THAT NO DATA ON DISK GOT CHANGED, BUT FIRST CLEAR ERROR

                10080   FILL READ INTO BUFFER WITH 0

                10087   FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ

                10094   FILL COMMAND

                10135   COMPARE READ DATA

                10155   SAVE REGISTERS FOR COMPARISON AFTER WRITE LOCK HAS BEEN
                        UNLOCKED

                10179   THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1

                10188   COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
                        WITH REGISTERS AFTER WRITE WAS UNLOCKED

                10216   TEST 51 END OF DRIVE
```

```
                10218           THIS IS THE END OF TEST FOR ONE DRIVE

                10220           IF THERE ARE MORE DRIVES, THEN THE PROGRAM
                                JUMPS TO TEST 4 FOR TESTING THE NEXT DRIVE

                10223           END PASS IS REACHED ONLY AFTER ALL DRIVES ARE TESTED

                ;*****************************************************************
        10266
                ;*****************************************************************

                ;*****************************************************************
        10267   ;*SUBROUTINES**
                ;*****************************************************************

                ;*****************************************************************
        10268
                ;*****************************************************************

                ;*****************************************************************
        10271   END OF PASS ROUTINE
                ;*****************************************************************

                10274   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

                ;*****************************************************************
        10312   JAM CURRENT CYLINDER ROUTINE
                ;*****************************************************************

                10314   THIS ROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER - 'RHCC'
                        BY GIVING A 'SEEK' COMMAND FOLLOWED BY AN INIT WHICH WILL LOAD
                        'RHCC' WITH THE DESIRED CYLINDER VALUE.  THE ROUTINE THEN CHECKS
                        THAT THE LOADED VALUE IS CORRECT.

                        CALL IS:
                                JSR     R0,@#MAKECYL
                                XC                              ;DESIRED VALUE OF CURRENT CYLINDER

                10363   THIS FILLS MEMORY WITH GIVEN DATA
                        USED CHIEFLY FOR HEADER INFORMATION
                        CALL IS
                                JSR     R0,@#FLHEAD     ;FILL HEADER
                                LOC                     ;LOCATION WHERE SAVED
                                XN                      ;NUMBER OF WORDS
                                XD1                     ;DATA REPEATED XN TIMES
                                XD2                     ;DATA REPEATED XN TIMES
```

```
10381   NOW FILL DATA

10392   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

10424   THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
        WITH GIVEN VALUE
        CALL IS
                JSR     R0,@#FILLRE
                RHXX            ;REGISTER NAME
                D               ;DATA


10445   THIS SUBROUTINE SETS UP FOR SEARCH
        CALL IS
                JSR     R0,@#SRCH
                C               ;CYLINDER
        .BYTE   S               ;SECTOR
        .BYTE   T               ;TRACK

10464   THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
        CALL IS
                JSR     R0,@#SEEKCY
                C               ;CYLINDER


10474   THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
        CALL IS
                JSR     R0,@#OFSET
                0               ;MICRO INCHES OFSET

10497   THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
        AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY
         (DRY) IN RHDS1 = 1

10501   IT ALSO CHECKS THAT THERE ARE NO BITS STUCK AT 1 IN RHDS1

10541   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

10580           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


10594           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

10627           THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS

10649           NOW TIME AND TOLERANCE WILL BE CHECKED

10695           THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
                NO TIMING IS DONE
                CALL IS
                        WAT
                        A       ;ABSOLUTE REGISTER ADDRESS
                        B       ;BIT WAITE) FOR
                        TA      ;TIME-NOT USED HERE
                        TO      ;TIME-NOT USED HERE
                R3-IS A TEMPORARY COUNTER

10707           THIS HAS THE TWO COUNT DOWNS FROM 177777

10730           BIT DID SET SO CHECK IF INTERRUPT OCCURED

10743           THIS CHANGES REGISTER SAVED VALUE
                CALL IS
                        JSR     R0,@#CHREG
                        R                       ;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

10773           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

10790   NOW DATA WILL BE FILLED

10806   THIS IS A SUBROUTINE TO COMPARE REGISTERS
        GOOD DATA IS ALREADY SAVED IN 'SAVERE'
        TEST DATA IS IN THE REGISTERS
        CALL IS
                JSR     R0,@#COMREG
                SAVERE                  ;GOOD DATA
                RHCS1                   ;ADDRESS OF ADDRESS TEST DATA
                N.                      ;RETURN FOR ERROR
                RG                      ;RETURN FOR GOOD COMPARISON
        ON RETURN WITH ERROR '$GODAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
        'REGADR' HAS REGISTER ADDRESS

10829   NOW SAVE REGISTERS

10933   NOW COMPARES WILL MADE

10961   HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
        ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
        PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

11065   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.

11128   THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
        IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"

11131   THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
        AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
        ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT

11155   THIS IS A DATA  COMMAND SETUP SUBROUTINE
        THE CALL IS
                JSR     R0,@#RUN
                C                       ;CYLINDER
        .BYTE   S                       ;SECTOR
        .BYTE   T                       ;TRACK
                -W                      ;WORD COUNT
                B                       ;BUS ADDRESS
```

```
                        BAI                     ;BUS ADDRESS INHIBIT
                        FMT22!ECI!HCI           ;FMT22=1 =16 BIT WORDS
                                                ;ECI = ECC CORRECTION INHIBIT
                                                ;HCI = HEADER COMPARE INHIBIT
                        COM                     ;COMMAND ADDRESS

        10983   THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY

        10985   R1 HAS GOOD DATA BUFFER ADDRESS
                R2 HAS TEST DATA BUFFER ADDRESS
                R5 HAS ADDRESS OF RETURN ON ERROR
                R3 HAS NUMBER OF WORDS TO BE COMPARED
                R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED

        10991   CALL IS:
                        JSR     R0,@#COMPAR
                        G                       ;ADDRESS OF GOOD DATA
                        T                       ;ADDRESS OF TEST DATA
                        N                       ;NUMBER OF WORDS TO BE COMPARED
                        RE                      ;RETURN ON ERROR
                        RG                      ;RETURN ON NO ERROR

        11038           THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                        ADDRESS FROM 176700 TO ANY TYPED VALUE

                ;***************************************************************
        11114   SCOPE HANDLER ROUTINE
                ;***************************************************************

        11117   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
```

```
                ;*****************************************************************
11177           ;CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                ;*****************************************************************

11180           ;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

                ;*****************************************************************
11244           ;TYPE ROUTINE
                ;*****************************************************************

11247           ;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


                ;*****************************************************************
11314           ;TTY INPUT ROUTINE
                ;*****************************************************************

11325           ;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

11342           ;TK SERVICE ROUTINE
                ;THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                ;BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                ;IT IN THE QUEUE.
                ;IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
                ;UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSE.
```

```
                11394   SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                        ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                        SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                        CALL WHEN OPERATING IN TTY INTERRUPT MODE.

                11409   CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                        ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                        CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

                11480   THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                        CALL:
                                RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                RETURN HERE             ;;CHARACTER IS ON THE STACK
                                                        ;;WITH PARITY BIT STRIPPED OFF

                11504   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

                        ********************************************************************
        11542   READ AN OCTAL NUMBER FROM THE TTY
                ********************************************************************

                11545   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

                        ********************************************************************
        11595   ERROR HANDLER ROUTINE
                ********************************************************************

                11598   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
```

```
                11644   ;**********************************

                        ;*********************************************************************
        11646   ERROR MESSAGE TYPEOUT ROUTINE
                ;*********************************************************************

                11648   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.
                        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.

                ;*********************************************************************
        11877   BINARY TO OCTAL (ASCII) AND TYPE
                ;*********************************************************************

                11880   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
```

```
                ;*********************************************************************
11954           ;TRAP DECODER
                ;*********************************************************************

11957           ;       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.

                ;*********************************************************************
11970           ;TRAP TABLE
                ;*********************************************************************

11972           ;       THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                ;       BY THE "TRAP" INSTRUCTION.

                ;*********************************************************************
11992           ;POWER DOWN AND UP ROUTINES
                ;*********************************************************************

12037           ;       ***********************************

                ;       ERROR AND MESSAGE TABLE CONDIMENTS

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

                ;*********************************************************************
                ;  40    57   168   178   189   208   245   303  1322  1505  1506  1507  1513  1525  161
                ;*********************************************************************
```

DZRJIA.P11 TABLE OF CONTENTS

41
42 **ERROR TABLE, BIT DEFINITIONS & STARTING ADDRESSES**
43
44 OPERATIONAL SWITCH SETTINGS
61 BASIC DEFINITIONS
172 TRAP CATCHER
182 ACT11 HOOKS
193 STARTING ADDRESSES
212 MEMORY MANAGEMENT DEFINITIONS
249 COMMON TAGS
307 ERROR POINTER TABLE
1326 REGISTER ADDRESSES
1509
1510 **DIAGNOSTIC CODE**
1511
1514 SETUP TESTS
1529 INITIALIZE THE COMMON TAGS
1616 GET VALUE FOR SOFTWARE SWITCH REGISTER
1679 T1 REFERENCE EACH REGISTER
1726 T2 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
1750 T3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
1897 T4 TYPE SERIAL NUMBER AND DRIVE TYPE
2059 T5 CHECK MOL TO BE HIGH
2082 T6 PROGRAM INTERRUPT
2134 T7 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
2175 T10 PACK ACKNOWLEDGE
2389 T11 SET VV BIT #6 IN RHDS1
2529 T12 MAKE CURRENT CYLINDER = 377
2549 T13 MAKE CURRENT CYLINDER = 777
2567 T14 ADDRESS PLUG CHANGE ERROR
2870 T15 CHECK ALL ADDRESS PLUG ADDRESSES
3036
3037 **DRIVE COMMAND TESTS**
3038
3045 T16 NO OPERATION FUNCTION TEST
3263 T17 DRIVE CLEAR
3479 T20 READ-IN-PRESET
3721 T21 READ-IN-PRESET
3903 T22 MAKE CURRENT CYLINDER = 777
3923 T23 MAKE CURRENT CYLINDER = 377
3943 T24 RECALIBRATE COMMAND
4128 T25 MAKE CURRENT CYLINDER = 777
4148 T26 MAKE CURRENT CYLINDER = 377
4167 T27 RECALIBRATE COMMAND
4314 T30 UNLOAD COMMAND
4673 T31 OFFSET AND RETURN TO CENTER LINE COMMAND
5019 T32 OFFSET COMMAND
5086 READ/WRITE TESTS USING MEDIA
5087 T33 WRITE/READ HEADER AND DATA (0'S)
5445 T34 READ DATA (0'S)
5603 T35 WRITE/READ DATA (1'S & 125252)
5966 T36 WRITE/READ DATA (125252)
6306 T37 WRITE/READ DATA (052525)
6673 T40 WRITE/READ DATA USING UNIBUS B

DZRJIA.P11 TABLE OF CONTENTS

 7078 T41 IMPLIED SEARCH
 7488 T42 IMPLIED SEEK TO CYL 001
 7492 SEEK TESTS
 7904 T43 SEEK & WRT TEST (CYL = 0-10)
 8551 T44 SEEK & READ TEST (CYL = 009)
 8810 WRITE CHECK DATA & WRITE PROTECT TESTS
 8812 T45 WRITE CHECK HEADER AND DATA
 9309 T46 WRITE CHECK DATA
 9557 T47 WRITE CHECK DATA USING UNIBUS B
 9827 T50 WRITE PROTECT OPERATION
10259 T51 END OF DRIVE
10311
10312 **SUBROUTINES**
10313
10316 END OF PASS ROUTINE
10357 JAM CURRENT CYLINDER ROUTINE
11159 SCOPE HANDLER ROUTINE
11223 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11290 TYPE ROUTINE
11360 TTY INPUT ROUTINE
11589 READ AN OCTAL NUMBER FROM THE TTY
11642 ERROR HANDLER ROUTINE
11693 ERROR MESSAGE TYPEOUT ROUTINE
11924 BINARY TO OCTAL (ASCII) AND TYPE
12001 TRAP DECODER
12017 TRAP TABLE
12039 POWER DOWN AND UP ROUTINES

```
                                .TITLE  MAINDEC-11-DZRJI-A, RP04/5/6 FUNCT. CONT. TST-PT 1
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY PETE BLACKSTONE
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-CO).MAR 21, 1976.
                                ;*

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


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

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

.SBTTL  OPERATIONAL SWITCH SETTINGS
;*
;*       SWITCH                  USE
;*       ------          --------------------
;*         15            HALT ON ERROR
;*         14            LOOP ON TEST
;*         13            INHIBIT ERROR TYPEOUTS
;*         12            RH70 CONTROLLER SELECT
;*         11            INHIBIT ITERATIONS
;*         10            BELL ON ERROR
;*          9            LOOP ON ERROR
;*          8            LOOP ON TEST IN SWR<7:0>
```

```
                                ;*          7           STOP FURTHER COMPARES IF SW08 IS LOW
                                ;*          6           TYPE ALL REG. WITH ERROR IF SW8 LOW
                                ;*          5           MULT ADDR PLUG TEST IF SW08 IS LOW
```

```
                                .SBTTL  BASIC DEFINITIONS

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

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

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

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
```

```
                    000010          SW03=   10
                    000004          SW02=   4
                    000002          SW01=   2
                    000001          SW00=   1
                                    .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
```

```
;65          000064          TPVEC=  64          ;;TTY PRINTER VECTOR
;66          000240          PIRQVEC=240         ;;PROGRAM INTERRUPT REQUEST VECTOR
;67
```

```
                                .SBTTL  TRAP CATCHER

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

                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
000200                                  $SVPC=.                 ;SAVE PC
000046                                  .=46
000046  041002                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                                  .=52
000052  002000                        .WORD   2000            ;;2)SET LOC.52 TO 2000
000200                                  .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESSES

000200                                  .=200
000200  000137  005012                JMP     @#BEGIN         ;NORMAL START
000210                                  .=210
000210  000137  004776                JMP     @#BEGIN2        ;JUMP TO SELECT DRIVE START
000220                                  .=220
000220  000137  004762                JMP     @#BEGIN1        ;JUMP TO NO OPERATOR TESTS START


                                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;*THIS WILL TEST ALL DRIVES ON THE SYSTEM A SINGLE DRIVE AT A TIME
                                ;*
                                ;*STARTING ADDRESS 210 WILL TEST,ONLY ONE SPECIFIED DRIVE
                                ;*
                                ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
                                ;*AT THE DRIVE
```

```
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                                ;*KT11 VECTOR ADDRESS

000250                          MMVEC=  250

                                ;*KT11 STATUS REGISTER ADDRESSES

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

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

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

                                ;*KERNEL "I" PAGE ADDRESS REGISTERS

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

                                ;;*******************************************************************
000110                                  .=110
```

```
                                .SBTTL  COMMON TAGS

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

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

F06


```
299  001222  077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
300  001223  015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
301  001224  000012       $LF:    .ASCII  <12>            ;;LINE FEED
302                        ;;**************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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

001226                          $ERRTB:



                                ;*ITEM1
001226  050570                          EM1             ;RP04 DID NOT INTERRUPT
                                                        ;WAITED ON BIT DID NOT OCCUR
001230  066642                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001232  071170                          DT1             ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001234  071476                          DF1             ;0,0,0,0,0,0

                                ;*ITEM2
001236  050617                          EM2             ;INTERRUPT ENABLE BIT DOWN BUT
                                                        ;WAITED ON BIT DID NOT OCCUR
001240  066642                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001242  071170                          DT1             ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001244  071476                          DF1             ;0,0,0,0,0,0

                                ;*ITEM3
001246  050706                          EM3             ;RP04 DID NOT INTERRUPT WHEN
                                                        ;WAITED ON BIT DID SET
001250  066642                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;RHCS1 CONTENTS
001252  071170                          DT1             ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001254  071476                          DF1             ;0,0,0,0,0,0
```

```
                                ;*ITEM4
001256  050767                          EM4     ;WAITED ON BIT DID SET BUT
                                                ;TIME IS IN ERROR
                                                ;TIME IS GIVEN IN 10 MICRO SEC.
                                                ;(DECIMAL)
001260  067021                          DH4     ;PC
                                                ;WAT PC
                                                ;BIT WAITED
                                                ;REG ADDRESS
                                                ;TIME IN 10 MSEC
001262  071210                          DT4     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,WAITIM
001264  071505                          DF4     ;0,0,0,0,0,1

                                ;*ITEM5
001266  051100                          EM5     ;RHAS DOES NOT CLEAR BY
                                                ;MOVING IN ALL ONES
001270  067162                          DH5     ;PC
                                                ;REG. ADDR.
                                                ;GOOD DATA
                                                ;RECEIVED DATA
001272  071230                          DT5     ;$ERRPC,REGADR,$GDDAT,$BDDAT
001274  071514                          DF5     ;0,0,0,0

                                ;*ITEM6
001276  051152                          EM6     ;LOADING RHER1 FOR ALL
                                                ;UNITS DID NOT SET ANY BITS
                                                ;IN RHAS-NO UNITS PRESENT
001300  067301                          DH6     ;PC
                                                ;REG ADDR
                                                ;RECEIVED DATA
001302  071244                          DT6     ;$ERRPC,REGADR,$BDDAT
001304  071521                          DF6     ;0,0,0

                                ;*ITEM7
001306  051240                          EM7     ;SPECIFIED REGISTER NONEXISTANT
                                                ;SO ABORT PROGRAM
001310  067400                          DH7     ;PC
                                                ;ADDR. OF REG.
001312  071256                          DT7     ;$ERRPC,TEMP1
001314  071525                          DF7     ;0,0

                                ;*ITEM10
001316  051322                          EM10    ;STOPED DRIVE HAS MOL BIT
                                                ;IN RHDS1 = 1
001320  067440                          DH10    ;PC
                                                ;TEST NO
                                                ;FAILING REG ADDR
                                                ;CONTENTS OF RHCS1
                                                ;CONTENTS OF RHCS2
                                                ;CONTENTS OF RHDS1
                                                ;CONTENTS OF RHER1
001322  071266                          DT10    ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001324  071530                          DF10    ;0,0,0,0,0,0,0
```

```
                                        ;*ITEM11
001326  051371                                EM11            ;WITH SPINDLE POWERED DOWN
                                                            ;RHCS2 SHOULD HAVE ONLY
                                                            ;UNIT NUMBER AND IR HIGH
001330  067440                                DH10            ;PC
                                                            ;TEST NO
                                                            ;FAILING REG. ADR
                                                            ;CONTENTS OF RHCS1
                                                            ;CONTENTS OF RHCS2
                                                            ;CONTENTS OF RHDS1
                                                            ;CONTENTS OF RHER1
001332  071266                                DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001334  071530                                DF10            ;0,0,0,0,0,0,0

                                        ;*ITEM12
001336  051476                                EM12            ;AFTER A POWER UP WITH
                                                            ;NO PACK ACKNOWLEDGE COMMAND
                                                            ;RHDS1 SHOULD HAVE MOL=1, VV=0
001340  067440                                DH10            ;PC
                                                            ;TEST NO
                                                            ;FAILING REGISTER ADDR.
                                                            ;CONTENTS OF RHCS1
                                                            ;CONTENTS OF RHCS2
                                                            ;CONTETNS OF RHDS1
                                                            ;CONTENTS OF RHER1
001342  071266                                DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001344  071530                                DF10            ;0,0,0,0,0,0,0

                                        ;*ITEM13
001346  051604                                EM13            ;AFTER A POWER UP WITHOUT
                                                            ;ANY INIT RHCS1 SHOULD
                                                            ;HAVE GO=0, DVA=1, RDY=1
                                                            ;IE=0, DISREGARD
                                                            ;ALL OTHER BITS
001350  067440                                DH10            ;PC
                                                            ;TEST NO
                                                            ;FAILING REGISTER ADDR.
                                                            ;CONTENTS OF RHCS1
                                                            ;CONTENTS OF RHCS2
                                                            ;CONTENTS OF RHDS1
                                                            ;CONTENTS OF RHER1
001352  071266                                DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001354  071530                                DF10            ;0,0,0,0,0,0,0

                                        ;*ITEM14
001356  051726                                EM14            ;AFTER POWER UP RHCC
                                                            ;SHOULD BE=0
001360  067162                                DH5             ;PC
                                                            ;REG. ADDR.
                                                            ;GOOD DATA
                                                            ;RECEIVED DATA
001362  071230                                DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001364  071514                                DF5             ;0,0,0,0
```

```
                                ;*ITEM15
001366  052001                          EM15            ;PACK ACKNOWLEDGE CAUSED
                                                        ;AN ERROR
                                                        ;GOOD DATA IS BEFORE COMMAND
                                                        ;RECEIVED DATA IS AFTER COMMAND
001370  067162                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001372  071230                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001374  071514                          DF5             ;0,0,0,0

                                ;*ITEM16
001376  052142                          EM16            ;GIVING A NO-OP COMMAND CAUSED
                                                        ;AN ERROR
                                                        ;GOOD DATA GIVES REGISTER
                                                        ;CONTENTS BEFORE COMMAND
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER COMMAND
001400  067162                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001402  071230                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001404  071514                          DF5             ;0,0,0,0

                                ;*ITEM17
001406  052270                          EM17            ;DRIVE CLEAR COMMAND
                                                        ;CAUSED AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
001410  067162                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001412  071230                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001414  071514                          DF5             ;0,0,0,0

                                ;*ITEM20
001416  052425                          EM20            ;READ-IN COMMAND GAVE AN ERROR
                                                        ;GOOD DATA HAS WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA HAS WHAT WAS
                                                        ;AFTER COMMAND
001420  067162                          DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001422  071230                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001424  071514                          DF5             ;0,0,0,0
```

```
001426  052571                          ;*ITEM 21
                                EM21                    ;RHCS1 CONTENTS DURING
                                                        ;COMMAND WAS IN ERROR
001430  067162                  DH5
001432  071230                  DT5
001434  071514                  DF5

                                ;*ITEM 22
001436  052644                  EM22                    ;RHDS1 CONTENTS DURING
                                                        ;COMM ANS WAS IN ERROR
001440  067162                  DH5
001442  071230                  DT5
001444  071514                  DF5

                                ;*ITEM 23
001446  052717                  EM23                    ;UNLOAD COMMAND GAVE AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
001450  067162                  DH5
001452  071230                  DT5
001454  071514                  DF5

                                ;*ITEM 24
001456  053062                  EM24                    ;OFFSET COMMAND CAUSED AN ERROR
                                                        ;GOOD DATA IS WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
                                                        ;AFTER AN OFFSET COMMAND
001460  067162                  DH5
001462  071230                  DT5
001464  071514                  DF5

                                ;*ITEM 25
001466  053225                  EM25                    ;RETURN TO CENTER LINE COMMAND
                                                        ;CAUSED AN ERROR
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND
001470  067162                  DH5
001472  071230                  DT5
001474  071514                  DF5

                                ;*ITEM 26
001476  053407                  EM26                    ;500 OFFSETS CAUSED AN ERROR
001500  067617                  DH26                    ;PC
                                                        ;CONT. OF RHCS1
                                                        ;CONT. OF RHCS2
                                                        ;CONT. OF RHDS1
                                                        ;CONT. OF RHER1
                                                        ;CONT. OF RHER2
                                                        ;CONT. OF RHER3
001502  071306                  DT26                    ;$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
```

```
001504  071537                          DF26            ;0,0,0,0,0,0,0
                                ;*ITEM 27
001506  053477                          EM27            ;WRITE HEADER AND DATA
                                                        ;CAUSED IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT
                                                        ;SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER COMMAND
001510  067162                          DH5
001512  071230                          DT5
001514  071514                          DF5
                                ;*ITEM 30
001516  053715                          EM30            ;WRITE HEADER AND DATA
                                                        ;CHANGED WRITE FROM BUFFER
001520  070016                          DH30            ;PC
                                                        ;WORD NO
                                                        ;GOOD DATA
                                                        ;BAD DATA
001522  071330                          DT30            ;$ERRPC,ERWORD,$GDDAT,$BDDAT
001524  071547                          DF30            ;0,0,0,0
                                ;*ITEM 31
001526  053775                          EM31            ;READ HEADER AND DATA CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA HAS WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER COMMAND
001530  067162                          DH5
001532  071230                          DT5
001534  071514                          DF5
                                ;*ITEM 32
001536  054212                          EM32            ;WRITE HEADER AND DATA FOLLOWED
                                                        ;BY A READ HEADER AND DATA
                                                        ;CAUSED A READ/WRITE ERROR
001540  070016                          DH30
001542  071330                          DT30
001544  071547                          DF30
                                ;*ITEM 33
001546  054321                          EM33            ;READ DATA CAUSED IMPROPER REGISTER
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
                                                        ;COMMAND
001550  067162                          DH5
001552  071230                          DT5
001554  071514                          DF5
                                ;*ITEM 34
001556  054523                          EM34            ;READ DATA INCORRECT
```

```
001560  070016                  DH30
001562  071330                  DT30
001564  071547                  DF30

                        ;*ITEM 35
001566  054547                  EM35            ;WRITE DATA COMMAND CAUSED
                                                ;IMPROPER REGISTER CHANGE
                                                ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                ;RECEIVED DATA GIVES REGISTER
                                                ;CONTENTS AFTER WRITE DATA
001570  067162                  DH5
001572  071230                  DT5
001574  071514                  DF5

                        ;*ITEM 36
001576  054765                  EM36            ;WRITE DATA COMMAND CHANGED
                                                ;WRITE FROM BUFFER
001600  070016                  DH30
001602  071330                  DT30
001604  071547                  DF30

                        ;*ITEM 37
001606  055042                  EM37            ;SEEK COMMAND CAUSED AN
                                                ;ERROR
                                                ;GOOD DATA GIVES WHAT SHOULD
                                                ;BE THERE
                                                ;RECEIVED DATA GIVES WHAT
                                                ;WAS THERE AFTER SEEK COMMAND
001610  067162                  DH5             ;
001612  071230                  DT5             ;
001614  071514                  DF5             ;

                        ;*ITEM 40
001616  055257                  EM40            ;WRITE CHECK CAUSED AN
                                                ;IMPROPER REGISTER CHANGE
                                                ;GOOD DATA GIVES WHAT SHOULD
                                                ;BE THERE
                                                ;RECEIVED DATA GIVES WHAT WAS
                                                ;THERE AFTER COMMAND
001620  067162                  DH5
001622  071230                  DT5
001624  071514                  DF5

                        ;*ITEM 41
001626  055466                  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
001630  067162                  DH5
```

```
001632  071230                          DF5
001634  071514                          DF5

                                ;*ITEM 42
001636  055747                          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
001640  067162                          DH5
001642  071230                          DT5
001644  071514                          DF5

                                ;*ITEM 43
001646  056225                          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
001650  070016                          DH30
001652  071330                          DT30
001654  071547                          DF30

                                ;*ITEM 44
001656  056567                          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
001660  067162                          DH5             ;
001662  071230                          DT5             ;
001664  071514                          DF5             ;

                                ;*ITEM 45
001666  057061                          EM45            ;TRANSFERRING ON LAST BLOCK IE. CYLINDER
                                                        ;410, SECTOR 21, TRACK 18
                                                        ;CAUSED IMPROPER REGISTER
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER TRANSFER
001670  067162                          DH5
001672  071230                          DT5
001674  071514                          DF5

                                ;*ITEM 46
```

```
001676  357355                  EM46            ;DATA READ FROM LAST
                                                ;BLOCK IE. CYLINDER 410
                                                ;SECTOR 21. TRACK 18 IS IN
                                                ;ERROR
001700  070016                  DM30
001702  071330                  DT30
001704  071547                  DF30

                        ;*ITEM 47
001706  357473                  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
001710  067162                  DM45
001712  071230                  DT45
001714  071514                  DF45

                        ;*ITEM 50
001716  357752                  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
001720  070016                  DM30
001722  071330                  DT30
001724  071547                  DF30

                        ;*ITEM 51
001726  360171                  EM51            ;GIVING ILLEGAL FUNCTION CAUSED
                                                ;IMPROPER REGISTER CHANGE
                                                ;GOOD DATA GIVES WHAT SHOULD BE
                                                ;THERE
                                                ;RECEIVED DATA GIVES REGISTER
                                                ;CONTENTS AFTER ILLEGAL FUNCTION
001730  070132                  DM51            ;PC
                                                ;REG. ADDR.
                                                ;GOOD DATA
                                                ;RECEIVED DATA
                                                ;ILLEGAL FUNCTION
001732  071344                  DT51            ;$ERRPC,REGADR,$GDDAT,$BDDAT,ILLEGL
001734  071554                  DF51            ;0,0,0,0,0

                        ;*ITEM 52
001736  360436                  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
001740  067162                        DH5
001742  071230                        DT5
001744  071514                        DF5

                                ;*ITEM 53
001746  060713                        EM53            ;READ HEADER AND DATA AFTER
                                                        ;A SEARCH CAUSED AN ERROR
001750  070016                        DH30
001752  071330                        DT30
001754  071547                        DF30

                                ;*ITEM 54
001756  061001                        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
001760  067162                        DH5
001762  071230                        DT5
001764  071514                        DF5

                                ;*ITEM 55
001766  061246                        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
001772  067162                        DH5
001774  071230                        DT5
001776  071514                        DF5

                                ;*ITEM 56
002000  061534                        EM56            ;DATA READ WITH AN EXPECTED
                                                        ;ADDRESS OVERFLOW ERROR IS
                                                        ;INCORRECT
                                                        ;WORD NO 1 TO 260 SHOULD
                                                        ;BE READ
                                                        ;WORD NOS 261 TO 266 SHOULD
                                                        ;NOT CHANGE DUE TO READ
002000  070016                        DH30
002002  071330                        DT30
002004  071547                        DF30

                                ;*ITEM 57
002006  061744                        EM57            ;ATTEMPTING DATA COMMAND
                                                        ;WITH WRONG FORMAT BIT CAUSED
                                                        ;IMPROPER REGISTER CHANGE
```

```
                                                                ;GOOD DATA GIVES WHAT SHOULD BE
                                                                ;THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER ATTEMPTED DATA
                                                                ;TRANSFER
002010  067162                    DH5
002012  071230                    DT5
002014  071514                    DF5

                        ;*ITEM 60
002016  062236                    EM60                          ;ATTEMPTING TO MODIFY REGISTER
                                                                ;DURING AN OPERATION CAUSED
                                                                ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER OPERATION
                                                                ;WAS COMPLETE
002020  070271                    D-60                          ;PC
                                                                ;REG. ADDR.
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA
                                                                ;MODFING REGISTER
002022  071362                    DT60                          ;$ERRPC,REGADR,$GDDAT,$BDDAT,$BCADR
002024  071562                    DF60                          ;0,0,0,0,0

                        ;*ITEM 61
002026  062647                    EM61                          ;DEVICE NOT AVAILBLE BEFORE COMMAND WAS TO BE GIVEN
002030  070426                    DH61                          ;PC
                                                                ;TEST NO.
                                                                ;PC OF JSR
                                                                ;RHCS1 CONTENTS
002032  071400                    DT61                          ;$ERRPC,$TSTNM,PCJSR,$BDADR
002034  071570                    DF61                          ;0,0,0,0

                        ;*ITEM 62
002036  062730                    EM62                          ;RHDS1 HAS STATUS BITS STUCK AT ONE
002040  070521                    DH62                          ;PC
                                                                ;TEST NO.
                                                                ;PC OF JSR
                                                                ;RHDS1 CONTENTS
002042  071412                    DT62                          ;$ERRPC,$TSTNM,PCJSR,$BDADR
002044  071574                    DF62                          ;0,0,0,0

                        ;*ITEM 63
002046  063011                    EM63                          ;RHDS1 CONTENTS DURING
                                                                ;COMMAND WAS IN ERROR
002050  067162                    DH5
002052  071230                    DT5
002054  071514                    DF5

                        ;*ITEM 64
```

```
                002056  063065                          EM64            ;RECALIBRATE COMMAND CAUSED
                                                                        ;IMPROPER REGISTER CHANGE.
                                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                                        ;THERE.
                                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
                                                                        ;AFTER COMMAND
                002060  067162                          DH5
                002062  071230                          DT5
                002064  071514                          DF5

                                        ;*ITEM65

                002066  063304                          EM65            ;INTERRUPT FAILING
                002070  070614                          DH65            ;PC
                                                                        ;TEST NO
                                                                        ;CONTENTS OF RHCS1
                                                                        ;CONTENTS OF RHAS
                                                                        ;CONTENTS OF RHDS1
                002072  071424                          DT65            ;$ERRPC,TSTNM,CS1,AS,DS1
                002074  071600                          DF65            ;0,0,0,0,0

                                        ;*ITEM66
                002076  063326                          EM66            ;HEADER AND DATA COMMAND
                                                                        ;FOR HEAD SELECTION TEST
                                                                        ;CAUSED AN ERROR
                                                                        ;RHDST GIVES WHAT TRACK
                                                                        ;WAS BEING WRITTEN ON CYLINDER 0
                                                                        ;SECTOR 0
                002100  070737                          DH66            ;PC
                                                                        ;RHDST
                                                                        ;RHER1
                                                                        ;RHER2
                                                                        ;RHER3
                                                                        ;RHCS1
                                                                        ;RHCS2
                002102  071440                          DT66            ;$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
                002104  071605                          DF66            ;0,0,0,0,0,0,0
                                        ;*ITEM67
                002106  063520                          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
                002110  070016                          DH30
                002112  071330                          DT30
                002114  071547                          DF30
                                        ;*ITEM70
                002116  064010                          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
002120  070016                    DH30
002122  071330                    DT30
002124  071547                    DF30

                        ;*ITEM 71
002126  064216                    EM71                 ;FORCING OPI CAUSED IMPROPER REGISTER
                                                        ; CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER 3 INDEX PULSES
002130  067162                    DH5                  ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
002132  071230                    DT5                  ;$ERRPC,REGADR,$GDDAT,$BDDAT
002134  071514                    DF5                  ;0,0,0,0
                        ;*ITEM 74
002136  064761                    EM74                 ;WHILE USING UNIBUS B
                                                        ;READ DATA CAUSED IMPROPER REGISTER
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
                                                        ;COMMAND
002140  067162                    DH5
002142  071230                    DT5
002144  071514                    DF5

                        ;*ITEM 73
002146  064707                    EM73                 ;WHILE USING UNIBUS B
                                                        ;READ DATA INCORRECT
002150  070016                    DH30
002152  071330                    DT30
002154  071547                    DF30

                        ;*ITEM 74
002156  064761                    EM74                 ;WHILE USING UNIBUS B
                                                        ;WRITE DATA COMMAND CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER WRITE DATA
002160  067162                    DH5
002162  071230                    DT5
002164  071514                    DF5

                        ;*ITEM 75
002166  065225                    EM75                 ;WHILE USING UNIBUS B
                                                        ;WRITE DATA COMMAND CHANGED
                                                        ;WRITE FROM BUFFER
```

```
1005  002170  070016                    DH30
1006  002172  071330                    DT30
1007  002174  071547                    DF30
1008
1009                              ;*ITEM 76
1010  002176  065330                    EM76        ;WHILE USING UNIBUS B
1011                                                  ;WRITE CHECK CAUSED AN
1012                                                  ;IMPROPER REGISTER CHANGE
1013                                                  ;GOOD DATA GIVES WHAT SHOULD
1014                                                  ;BE THERE
1015                                                  ;RECEIVED DATA GIVES WHAT WAS
1016                                                  ;THERE AFTER COMMAND
1017  002200  067162                    DH5
1018  002202  071230                    DT5
1019  002204  071514                    DF5
1020
1021                              ;*ITEM 77
1022  002206  065565                    EM77        ;CURRENT CYLINDER DOES NOT REFLECT DESIRED 'RHCC'
1023  002210  071035                    DH77        ;PC
1024                                                  ;PC OF JSR
1025                                                  ;REGISTER ADDRESS
1026                                                  ;GOOD DATA
1027                                                  ;BAD DATA
1028  002212  071460                    DT77        ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
1029  002214  071615                    DF77        ;0,0,0,0,0
1030
1031                              ;*ITEM 100
1032  002216  066010                    EM100       ;ERROR AFTER ADDRESS PLUG CHANGE
1033  002220  067162                    DH5         ;PC
1034                                                  ;REGISTER ADDRESS
1035                                                  ;GOOD DATA
1036                                                  ;RECEIVED DATA
1037  002222  071230                    DT5         ;$ERRPC,REGADR,$GDDAT,$BDDAT
1038  002224  071514                    DF5         ;0,0,0,0
1039
1040                              ;*ITEM 101
1041  002226  066072                    EM101       ;UNIT DID NOT GO OFFLINE WHEN ADDR
1042                                                  ;PLUG WAS REMOVED
1043  002230  067617                    DH26        ;PC
1044                                                  ;CONT OF RHCS1
1045                                                  ;CONT OF RHCS2
1046                                                  ;CONT OF RHDS1
1047                                                  ;CONT OF RHER2
1048                                                  ;CONT OF RHER3
1049  002232  071306                    DT26        ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1050  002234  071537                    DF26        ;0,0,0,0,0,0,0
1051
1052                              ;*ITEM 102
1053  002236  066154                    EM102       ;UNIT DID NOT COME BACK ONLINE WHEN
1054                                                  ;ADDR PLUG WAS REPLACED
1055  002240  067617                    DH26        ;PC
1056                                                  ;CONT OF RHCS1
1057                                                  ;CONT OF RHCS2
1058                                                  ;CONT OF RHDS1
```

```
1059
1060
1061  002242  071306                    DT26        ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1062  002244  071537                    DF26        ;0,0,0,0,0,0,0
1063
1064                          :*ITEM  103
1065  002246  066233                    EM103       ;REGISTER CONTENTS INCORRECT BEFORE A
1066                                                  ;DIAGNOSTIC SEEK
1067  002250  067617                    DH26        ;PC
1068                                                  ;CONT OF RHCS1
1069                                                  ;CONT OF RHCS2
1070                                                  ;CONT OF RHDS1
1071                                                  ;CONT OF RHER2
1072                                                  ;CONT OF RHER3
1073  002252  071306                    DT26        ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1074  002254  071537                    DF26        ;0,0,0,0,0,0,0
1075
1076                          :*ITEM  104
1077  002256  066317                    EM104       ;REGISTER CONTENTS INCORRECT AFTER A
1078                                                  ;DIAGNOSTIC SEEK
1079  002260  067617                    DH26        ;PC
1080                                                  ;CONT OF RHCS1
1081                                                  ;CONT OF RHCS2
1082                                                  ;CONT OF RHDS1
1083                                                  ;CONT OF RHER2
1084                                                  ;CONT OF RHER3
1085  002262  071306                    DT26        ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1086  002264  071537                    DF26        ;0,0,0,0,0,0,0
1087
```

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


002266  000254           RPVEC:  254                     ;RP04 VECTOR ADDRESS


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


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


                        ;*CONTROL AND STATUS REGISTER 2 (RHCS2)

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

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

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



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

000001                          GO=     1               ;GO (BIT #0)
000100                          IE=     100             ;INTERRUPT ENABLE (BIT #6)
000200                          RDY=    200             ;READY (BIT #7)
000400                          A16=    400             ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                          A17=    1000            ;HIGH ORDER UNIBUS BITS (BIT #9)
002000                          PSEL=   2000            ;PORT SELECT (BIT #10)
004000                          DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
020000                          MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
040000                          TRE=    40000           ;TRANSFER ERROR (BIT #14)
100000                          SC=     100000          ;SPECIAL CONDITION (BIT #15)

                                ;*STATUS REGISTER (RHDS1) (#01)

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

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

```
1185            100000                  DCK=    100000                  ;DATA CHECK ERROR (BIT 15)
1186
1187                                    ;*MAINTAINABILITY REGISTER (RHMR)(#03)
1188
1189            000001                  DMD=    1                       ;DIAGINOSTIC MODE (BIT #0)
1190            000002                  MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
1191            000004                  MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
1192            000010                  MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1193            000020                  MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
1194            000040                  MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
1195            001000                  DTSY=   1000                    ;MAINTAINABILITY SYNC DETECTED (BIT #9)
1196
1197                                    ;*ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1198
1199            000001                  AT0=    1                       ;DEVICE 0 (BIT #0)
1200            000002                  AT1=    2                       ;DEVICE 1 (BIT #1)
1201            000004                  AT2=    4                       ;DEVICE 2 (BIT #2)
1202            000010                  AT3=    10                      ;DEVICE 3 (BIT #3)
1203            000020                  AT4=    20                      ;DEVICE 4 (BIT #4)
1204            000040                  AT5=    40                      ;DEVICE 5 (BIT #5)
1205            000100                  AT6=    100                     ;DEVICE 6 (BIT #6)
1206            000200                  AT7=    200                     ;DEVICE 7 (BIT #7)
1207
1208                                    ;*DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1209                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1210
1211
1212
1213
1214
1215                                    ;*DRIVE TYPE REGISTER (RHDT) (#06)
1216                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1217
1218
1219
1220
1221
1222                                    ;*LOCK-AHEAD REGISTER (RHLA) (#07)
1223
1224            000001                  EXT1=   1                       ;EXTENSION 1 (BIT #0)
1225            000002                  EXT2=   2                       ;EXTENSION 2 (BIT #1)
1226            000004                  EXT4=   4                       ;EXTENSION 3 (BIT #2)
1227            000010                  EXT10=  10                      ;EXTENSION 4 (BIT #3)
1228            000020                  EXT20=  20                      ;EXTENSION 5 (BIT #4)
1229            000040                  EXT40=  40                      ;EXTENSION 6 (BIT #5)
1230            000100                  SC1=    100                     ;SECTOR COUNT FIELD 0 (BIT #6)
1231            000200                  SC2=    200                     ;SECTOR COUNT FIELD 1 (BIT #7)
1232            000400                  SC4=    400                     ;SECTOR COUNT FIELD 2 (BIT #8)
1233            001000                  SC10=   1000                    ;SECTOR COUNT FIELD 3 (BIT #9)
1234            002000                  SC20=   2000                    ;SECTOR COUNT FIELD 4 (BIT #10)
1235            004000                  TRK1=   4000                    ;TRACK FIELD 1 (BIT #11)
1236            010000                  TRK2=   10000                   ;TRACK FIELD 2 (BIT #12)
1237            020000                  TRK4=   20000                   ;TRACK FIELD 3 (BIT #13)
1238            040000                  TRK10=  40000                   ;TRACK FIELD 4 (BIT #14)
```

```
1239            100000          TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)
1240
1241                            ;*ERROR REGISTER #2 (RHER2) (#10)
1242
1243            000001          WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
1244            000002          CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
1245            000004          WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
1246            000010          CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
1247            000020          MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
1248            000040          TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1249            000100          TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
1250            000200          FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
1251            000400          WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
1252            001000          MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
1253            002000          NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
1254            004000          IXE=    4000                    ;INDEX ERROR (BIT #11)
1255            010000          VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
1256            020000          PLU=    20000                   ;PLO UNSAFE (BIT #13)
1257            100000          ACU=    100000                  ;ACUNSAFE (BIT #15)
1258
1259                            ;*OFFSET REGISTER (RHOF) (#11)
1260
1261            000001          OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
1262            000002          OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
1263            000004          OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
1264            000010          OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
1265            000020          OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
1266            000040          OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
1267
1268            000200          OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1269            002000          HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
1270            004000          ECI=    4000                    ;EPROR CORRECTION CODE INHIBIT (BIT #11)
1271            010000          FMT22=  10000                   ;FORMAT BIT (BIT #12
1272
1273                            ;*DESIRED CYLINDER ADDRESS (RHCA) (#12)
1274                            ;*EACH BIT IS CALLED BY BIT NUMBER.
1275
1276
1277
1278
1279                            ;*CURRENT CYLINDER ADDRESS (RHCC) (#13)
1280                            ;*EACH BIT IS CALLED BY BIT NUMBER
1281
1282
1283
1284
1285                            ;*SERIAL NUMBER REGISTER (RHSN) (#14)
1286                            ;*EACH IS CALLED BY BIT NUMBER
1287
1288
1289
1290
1291                            ;*ERROR REGISTER #03 (RHER3) (#15)
1292
```

```
1293            000001                  PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
1294            000002                  VUF=    2               ;VELOCITY UNSAFE (BIT #1)
1295            000010                  UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1296            000020                  PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
1297            000040                  ACL=    40              ;AC LOW (BIT #5)
1298            000100                  DCL=    100             ;DC LOW (BIT #6)
1299            020000                  ACE=    20000           ;ADDRESS CHANGE ERROR (BIT #13)
1300            040000                  SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
1301            100000                  OCYL=   100000          ;OFF CYLINDER (BIT #15)
1302
1303
1304
1305                                    ;*ECC POSITION REGISTER (RHEC1) (#16)
1306                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1307
1308
1309
1310
1311                                    ;*ECC PATTERN REGISTER (RHEC2) (#17)
1312                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1313
1314
1315
1316
1317
1318
1319
1320
1321
```

```
1322                                    .SBTTL  REGISTER ADDRESSES
1323
1324
1325
1326
1327                                    ;*RP04/5/6 DISK I/O REGISTER LOCATED IN THE RH11 CONTROLLER
1328
1329  002270  176722                    RHDB:   176722          ;DATA BUFFER
1330  002272  176702                    RHWC:   176702          ;WORD COUNT
1331  002274  176704                    RHBA:   176704          ;BUS ADDRESS
1332  002276  176710                    RHCS2:  176710          ;CONTROL AND STATUS 2
1333
1334                                    ;*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)
1335
1336  002300  176700                    RHCS1:  176700          ;CONTROL AND STATUS 1
1337  002302  176714                    RHER1:  176714          ;ERROR #1
1338  002304  176706                    RHDST:  176706          ;DESIRED SECTOR/TRACK ADDRESS
1339  002306  176740                    RHER2:  176740          ;ERROR #2
1340  002310  176732                    RHOF:   176732          ;OFFSET
1341  002312  176734                    RHCA:   176734          ;DESIRED CYLINDER ADDRESS
1342  002314  176742                    RHER3:  176742          ;ERROR #3
1343  002316  176716                    RHAS:   176716          ;ATTENTION SUMMARY
1344  002320  176724                    RHMR:   176724          ;MAINTAINABILITY
1345  002322  176712                    RHDS1:  176712          ;DRIVE STATUS
1346  002324  176726                    RHDT:   176726          ;DRIVE TYPE
1347  002326  176730                    RHSN:   176730          ;SERIAL NUMBER
1348  002330  176744                    RHEC1:  176744          ;ECC POSITION
1349  002332  176746                    RHEC2:  176746          ;ECC PATTERN
1350  002334  176736                    RHCC:   176736          ;CURRENT CYLINDER ADDRESS
1351  002336  176720                    RHLA:   176720          ;LOOK-AHEAD
1352
1353                                    ;*ADDITIONAL I/O REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC
1354
1355  002340  176750                    RHBAE:  176750          ;BUS ADDRESS EXTENSION REGISTER
1356  002342  176752                    RHCS3:  176752          ;CONTROL AND STATUS REGISTER #3
1357
1358
1359                                    ;*P-CLOCK (KW11-P) I/O REGISTERS
1360
1361  002344  172540                    PCLCSR: 172540          ;CONTROL AND STATUS REGISTER
1362  002346  172542                    PCLBUF: 172542          ;COUNT SET BUFFER
1363  002350  172544                    PCLCTR: 172544          ;COUNTER
1364
```

```
                        ;*THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SNAPSHOTS
                        ;*ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED

                        ;*ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
                        ;*FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND


                        ;;*****************************************************************
                        ;*THIS ASSUMES THAT A REGISTER SNAPSHOT HAS BEEN TAKEN WHICH IS NOT
                        ;*ALWAYS THE CASE - IF QUESTIONABLE CONTENTS APPEAR IN THE REGISTER
                        ;*PRINTOUTS, CHECK THE INLINE TEST CODE TO SEE IF THE REGISTER SNAPSHOT
                        ;*REFLECTS THE CURRENT STATE OF THE MACHINE
                        ;;*****************************************************************

 002352  000000   DB:    0          ;DATA BUFFER
 002354  000000   WC:    0          ;WORD COUNT
 002356  000000   BA:    0          ;BUS ADDRESS
 002360  000000   CS2:   0          ;CONTROL AND STATUS 2

 002362  000000   CS1:   0          ;CONTROL AND STATUS 1
 002364  000000   ER1:   0          ;ERROR #1
 002366  000000   DST:   0          ;DESIRED SECTOR/TRACK ADDRESS
 002370  000000   ER2:   0          ;ERROR #2
 002372  000000   OF:    0          ;OFFSET
 002374  000000   DCA:   0          ;DESIRED CYLINDER ADDRESS
 002376  000000   ER3:   0          ;ERROR #3
 002400  000000   AS:    0          ;ATTENTION SUMMARY
 002402  000000   MR:    0          ;MAINTAINABILITY
 002404  000000   DS1:   0          ;DRIVE STATUS
 002406  000003   DT:    0          ;DRIVE TYPE
 002410  000000   SN:    0          ;SERIAL NUMBER
 002412  000000   ECPS:  0          ;ECC POSITION
 002414  000000   ECPT:  0          ;ECC PATTERN
 002416  000000   CCA:   0          ;CURRENT CYLINDER ADDRESS
 002420  000000   LA:    0          ;LOOK-AHEAD
```

```
                                        ;*FUNCTION EQUATES

                                            ;*TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
002422                                  FUTABL:
002422  000000                          NOPERA: 0                       ;NO OPERATION
002424  000002                          UNLOAD: 2                       ;UNLOAD (STAND BY)
002426  000006                          RECALI: 6                       ;RECALIBRATE
002430  000010                          DCLEAR: 10                      ;DRIVE CLEAR
002432  000012                          RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
002434  000030                          SERCH:  30                      ;SEARCH COMMAND
002436  000050                          WRCHEK: 50                      ;WRITE CHECK DATA
002440  000052                          WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
002442  000060                          WRIDAT: 60                      ;WRITE DATA
002444  000062                          WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT
002446  000070                          READAT: 70                      ;READ DATA
002450  000072                          REFOR:  72                      ;READ HEADER AND DATA
002452  000004                          SEECOM: 4                       ;SEEK COMMAND
002454  000014                          OFSETC: 14                      ;OFFSET COMMAND
002456  000016                          RETCL:  16                      ;RETURNTO CENTERLINE
002460  000022                          PKACK:  22                      ;PACK ACKNOWLEDGE
002462  000020                          READIN: 20                      ;READ IN
002464  000000                          ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION



                                        ;*DATA BUFFERS FOR READ WRITE

002470  000422                          WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
003534  000422                          REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
```

```
                                        ;*RESERVED CORE LOCATIONS

004600  000000                          REGADR: 0                       ;SAVE REGISTER ADDRESS HERE
004602  000000                          ERWORD: 0                       ;SAVE ERROR WORD NUMBER HERE
004604  000000                          TSTNM:  0                       ;TEST NUMBER
004606  000000                          RP4VEC: 0                       ;CONTAINS ADDRESS OF LOCATION
                                                                        ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
                                                                        ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
                                                                        ;EFFECTIVE.

004610  000000                          OFSTVL: 0                       ;OFFSET VALUE USED IN OFFSET TEST

004612  000024                          SAVERE: .BLKW   20.             ;BLOCK TO SAVE REGISTERS FOR PRETEST
                                                                        ;HARDWARE REGISTER SNAPSHOTS - THESE
                                                                        ;ARE USUALLY THEN CHANGED TO REFLECT
                                                                        ;EXPECTED CONDITIONS AFTER THE TEST
004662  000000                          FINALA: 0                       ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
004664  000000                          FINACC: 0                       ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION

                                        ;*TABLE FOR ATTENTION BITS
                                        ;*ATTENTION TABLE

004666      001     002     004 ATABLE: .BYTE   1,2,4,10,20,40,100,200
004670      010     020     040
004674      100     200
```

```
1468
1469
1470                                    ;*FLAGS AND INTERNAL PROGRAM CONTROL WORDS
1471
1472  004676  000010                    UNITS:  .BLKW   8.              ;*THIS IS FILLED WITH -1
1473  004716  000000                    UNIT:   .WORD   0               ;UNIT UNDER TEST
1474  004720  000000                    NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
1475                                                                    ;USED TO KEEP TRACK OF UNIT UNDER TEST
1476  004722  000000                    NUNIT:  .WORD   0               ;USED TO DETERMIN IF THERE ARE MORE
1477                                                                    ;THAN ONE UNIT
1478  004724  000000                    NOPUSH: 0                       ;ALL ONES INDICATE NONE OF THE OPERATOR
1479                                                                    ;INTERVENTION TESTS WILL BE PERFORMED
1480  004726  000000                    SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
1481  004730  000000                    UNITSL: .WORD   0               ;UNIT NO. SELECTED
1482  004732  000000                    UBUSB:  0                       ;IF ZERO UNIBUS PRESENT
1483                                                                    ;IF ONES NO UNIBUS 8
1484  004734  000000                    ERFLG$: 0                       ;ERROR FLAG
1485  004736  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
1486                                                                    ;IF ONES WILL NOT TYPE HEADER
1487
1488  004740  000000                    ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
1489  004742  000000                    TOTALAT:0                       ;TOTAL ATTENTION BITS
1490
1491  004744  000000                    RP06:   0                       ;RP06 DEVICE TYPE FLAG LOCATION
1492  004746  000000                    RP05:   0                       ;MEMOREX RP04 DEVICE TYPE FLAG
1493  004750  000000                    RH70:   0                       ;IF = 1, PROGRAM IS RUNNING ON RWP04 SYSTEM
1494                                                                    ;IF = 0, PROGRAM IS RUNNING ON RJP04
1495  004752  000000                    INUNIT: 0                       ;INITIAL UNIT NO. - USED DURING
1496                                                                    ;CHECKING ALL ADDRESS PLUG ADDRESSES
1497
1498
1499
1500
1501
1502  004754  000000                    TMP0:   .WORD   0               ;TEMP STORAGE
1503  004756  000000                    TMP1:   .WORD   0
1504  004760  000000                    TMP4:   .WORD   0               ;TEMP STORAGE
```

```
DZRJIA.P11      WORD    0               ;TEMP STORAGE

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


                                                .SBTTL  SETUP TESTS


005762  012737  177777  004724  BEGIN1: MOV     #-1,@#NOPUSH    ;JUMP OVER OPERATOR REQUIRED TESTS
005770  005037  004726                  CLR     @#SELECT        ;DO NOT SELECT UNIT
005774  000412                          BR      START
005776  012737  177777  004726  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
006004  005037  004724                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
006010  000404                          BR      START
006012  005037  004726          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
006016  005037  004724                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY OPERATOR
                                                                ;INTERVENTION TESTS - NORMAL RUN

006022                          START:
006022  000005                          RESET
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
006024  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
006030  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
006032  022706  001140                  CMP     #SWR,R6 ;;DONE?
006036  001374                          BNE     .-6             ;;LOOP BACK IF NO
006040  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
006044  012737  044414  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
006052  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
006060  012737  046612  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
006066  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
006074  012737  050336  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
006102  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
006110  012737  050406  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
006116  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
006124  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
006130  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
006134  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
006142  012737  005142  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
006150  012737  005150  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
006156  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
006162  012737  005216  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
006170  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
006176  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
006204  022777  177777  173726          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
006212  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
006214  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
006216  012716  005224          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
006222  000002                          RTI
006224  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
```

DZRJIA.P11 INITIALIZE THE COMMON TAGS

```
1559  005232  012737  000174  001142          MOV     #DISPREG,DISPLAY
1560  005240  012637  000304          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1561
1562
1563
1564
1565  005244  012737  000000  177776          MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
1566  005252  012737  000200  000036          MOV     #200,@#TRAPVEC+2;TRAP PRIORITY = 4
1567  005260  013700  002266                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
1568  005264  012720  044352                  MOV     #RPVECT,(R0)+   ;THIS IS FOR UNTIMELY INTERRUPTS
1569  005270  012710  000340                  MOV     #340,(R0)       ;DRIVE INTERRUPT SERVICE ROUTINE
1570                                                                  ;PRIORITY = 7
1571
1572  005274  004737  045352                  JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
1573  005300  005737  004736                  TST     @#FIRST         ;IS THIS FIRST TIME ROUND ?
1574  005304  001001                          BNE     1$              ;DON'T GIVE HEADER IF NOT
1575  005306  000402                          BR      2$              ;HEADER 1ST TIME THROUGH
1576  005310  000137  006110          1$:     JMP     @#SND1          ;NO HEADER
1577
1578  005314                          2$:
1579  005314  104400  005322                  TYPE    ,68$            ;;TYPE ASCIZ STRING
1580  005320  000435                          BR      67$             ;;GET OVER THE ASCIZ
1581                                  ;;68$:  .ASCIZ  <15><12>?RP04/5/6 FUNCTIONAL CONTROLLER TEST - PART I  - DZRJI-A?
1582  005414                          67$:
1583  005414  104400  005422                  TYPE    ,70$            ;;TYPE ASCIZ STRING
1584  005420  000417                          BR      69$             ;;GET OVER THE ASCIZ
1585                                  ;;70$:  .ASCIZ  <15><12>/REVISION DATE: 21-MAR-76/<15><12>
1586  005460                          69$:
1587
1588  005460  104400  005466                  TYPE    ,72$            ;;TYPE ASCIZ STRING
1589  005464  000435                          BR      71$             ;;GET OVER THE ASCIZ
1590                                  ;;72$:  .ASCIZ  <15><12>/ALL DCL'S UNDER TEST MUST BE LOCKED ON THE CORRECT PORT/
1591  005560                          71$:
1592  005560  104400  005566                  TYPE    ,74$            ;;TYPE ASCIZ STRING
1593  005564  000433                          BR      73$             ;;GET OVER THE ASCIZ
1594                                  ;;74$:  .ASCIZ  <15><12>/IF CHANGE IS REQUIRED ON PORT SWITCH, THEN A CYCLE/
1595  005654                          73$:
1596  005654  104400  005662                  TYPE    ,76$            ;;TYPE ASCIZ STRING
1597  005660  000435                          BR      75$             ;;GET OVER THE ASCIZ
1598                                  ;;76$:  .ASCIZ  <15><12>/UP SEQUENCE IS REQ FOR STROBING THE PORT SELECT FLOP/<15><12>
1599  005754                          75$:
1600
1601  005754  104400  005762                  TYPE    ,78$            ;;TYPE ASCIZ STRING
1602  005760  000426                          BR      77$             ;;GET OVER THE ASCIZ
1603                                  ;;78$:  .ASCIZ  <15><12>/ALL DCL'S NOT UNDER TEST MUST BE SWITCHED
1604  006036                          77$:
1605  006036  104400  006044                  TYPE    ,80$            ;;TYPE ASCIZ STRING
1606  006042  000422                          BR      79$             ;;GET OVER THE ASCIZ
1607                                  ;;80$:  .ASCIZ  <15><12>/OFF, OR LOCKED ON THE OTHER PORT
1608  006110                          79$:
1609
1610  006110  012737  177777  004736  SND1:   MOV     #-1,@#FIRST     ;NEXT TIME DO NOT GIVE HEADER
1611
1612                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
```

H08


```
1613  006116  005737  000042                    TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1614  006122  001006                            BNE     64$             ;;BRANCH IF YES
1615  006124  023727  001140  000176            CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1616  006132  001005                            BNE     65$             ;;BRANCH IF NO
1617  006134  104405                            GTSWR                   ;;GET SOFT-SWR SETTINGS
1618  006136  000403                            BR      65$
1619  006140  112737  000001  001134  64$:     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1620  006146                          65$:
1621
1622  006146  032777  010000  172764  RH70CK:  BIT     #SW12,@SWR      ;LOOK TO SEE IF USING RH70
1623  006154  001403                            BEQ     3$              ;IF SW12 = 0, SKIP NEXT
1624  006156  012737  177777  000750            MOV     #-1,@#RH70      ;IF SW12 = 1, CU IS AN RH70
1625  006164                          3$:
```

```
1626                                    ;;*****************************************************************
1627                                    ;*IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
1628                                    ;*IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
1629                                    ;*IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
1630                                    ;*IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
1631                                    ;*IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'
1632
1633                                    ;*THE NEXT LINE IS TO BE ADDED LATER
1634                                    ;*AND THE JUMP AND NOP REMOVED
1635                                    ;*FOR NOW NO CLOCK WILL BE USED
1636
1637                                    ;       MOV     @#1$,@#ERRVEC   ;SET TIME-OUT VECTOR
1638                                    ;
1639                                    ;       JMP     @#1$            ;DO NOT USE CLOCK
1640                                    ;       NOP
1641                                    ;       TST     @#PCLCSR        ;REFERENCE P-CLOCK STATUS REGISTER
1642                                    ;                               ;ADDRESS = 172540
1643                                    ;       MOV     #WAIT.P,@#$TRPAD+20 ;THERE IS A P-CLOCK
1644                                    ;       MOV     #TIME1,@#RP4VEC ;THERE IS A P CLOCK SO
1645                                    ;                               ;VECTOR TO TIME1
1646                                    ;       BR      2$
1647                                    ;1$:    MOV     #WAIT.T,@#$TRPAD+20 ;THERE IS NO P-CLOCK
1648                                    ;;*****************************************************************
1649
1650
1651
1652  006164  012737  041600  004606          MOV     #TIME2,@#RP4VEC ;RP04/5/6 INTERRUPTS GO TO 'TIME 2'
1653  006172  012737  177777  046760  2$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
1654
1655
1656
1657
1658  006200  005737  004726                  TST     @#SELECT        ;WAS IT A 200 START
1659  006204  001442                          BEQ     TST1    ;       BRANCH IF STARTING FROM 200
1660  006206  104400  006214                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1661  006212  000424                          BR      64$             ;;GET OVER THE ASCIZ
1662                                  ;;65$:  .ASCIZ  <15><12>/SELECT UNIT NUMBER TO BE TESTED ?   <15><12>
1663  006264                          64$:
1664  006264  104411                          RDOCT
1665  006266  042716  177770                  BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
1666  006272  011637  004716                  MOV     (SP),@#UNIT     ;SAVE UNIT TO BE TESTED
1667  006276  012637  004730                  MOV     (SP)+,@#UNITSL  ;SAVE UNIT TO BE TESTED
1668
1669
1670
1671  006302  001403                          BEQ     TST1    ;       BRANCH IF STARTING FROM 200
1672  006304  013737  004730  004716          MOV     @#UNITSL,@#UNIT ;SET UNIT NUMBER
1673
```

```
1674
1675                                    ;;*****************************************************************
1676                                    ;*TEST 1         REFERENCE EACH REGISTER
1677
1678                                    ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
1679
1680                                    ;;*****************************************************************
1681  006312  000004                    TST1:   SCOPE
1682  006314  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
1683  006322  012737  000001  004604            MOV     #2-1,@#TSTNM    ;THIS SAVES TEST NUMBER
1684  006330  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
1685  006334  012737  046622  000030            MOV     #REGSAV,@#EMTVEC;ERROR VECTOR SO THAT
1686                                                                    ;NO REGISTERS ARE SAVED
1687  006342  012737  006370  000004            MOV     #2$,@#ERRVEC    ;SET UP FOR BUS TIMEOUT
1688
1689  006350  012700  000024                    MOV     #24,R0          ;THERE ARE 24 REG TO TEST
1690  006354  012701  002270                    MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
1691  006360  013102                    1$:     MOV     @(R1)+,R2       ;READ HARDWARE REG.
1692  006362  005300                            DEC     R0              ;COUNT DOWN
1693  006364  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
1694  006366  000470                            BR      3$              ;BRANCH IF 24 DONE
1695  006370  012737  000006  000004  2$:      MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1696  006376  022626                            CMP     (SP)+,(SP)+     ;CLEAN STACK
1697  006400  016137  177776  001200            MOV     -2(R1),$TMP1    ;STORE FAILING REG ADDR
1698  006406  104007                            ERROR   7               ;REGISTER NON EXISTANT
1699  006410  032777  020000  172522            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
1700  006416  001052                            BNE     4$              ;BRANCH IF YES
1701
1702  006420  104400  006426                    TYPE    ,65$            ;;TYPE ASCIZ STRING
1703  006424  000424                            BR      64$             ;;GET OVER THE ASCIZ
1704                                    ;;65$:  .ASCIZ  <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT
1705  006476                            64$:
1706  006476  104400  006504                    TYPE    ,67$            ;;TYPE ASCIZ STRING
1707  006502  000415                            BR      66$             ;;GET OVER THE ASCIZ
1708                                    ;;67$:  .ASCIZ  <15><12>/PROGRAM AND RESTART AT /
1709  006536                            66$:
1710
1711  006536  012746  043522                    MOV     #BASECH,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
1712                                                                    ;OF "CHANGE OF BASE ADDRESS" ROUTINE
1713  006542  104401                            TYPOC
1714  006544  000137  040714            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM ---------->
1715
1716  006550  012737  046612  000030  3$:      MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
1717                                                                    ;SO THAT REGISTERS ARE SAVED
1718  006556  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
```

```
1719
1720
1721                                    ;;*************************************************************
1722                                    ;*TEST 2        PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
1723
1724                                    ;*      CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES INTO IT
1725
1726                                    ;;*************************************************************
1727  006564  000004                   TST2:   SCOPE
1728  006566  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
1729  006574  012706  001000           MOV     #STACK,SP       ;SET STACK POINTER
1730  006600  012737  000002  004604           MOV     #3-1,@#TSTNM    ;THIS SAVES TEST NUMBER
1731
1732  006606  013701  002316           MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
1733  006612  012711  177777           MOV     #-1,@R1         ;THIS WRITES ALL ONES INTO RHAS
1734  006616  105711                   TSTB    @R1             ;TEST RHAS FOR ALL 0'S
1735  006620  001407                   BEQ     TST3    ;      BRANCH IF GOOD
1736  006622  011137  001126           MOV     @R1,@#$BDDAT    ;BAD DATA
1737  006626  005037  001124           CLR     @#$GDDAT        ;GOOD DATA
1738  006632  010137  004600           MOV     R1,@#REGADR     ;FAILING REG. (RHAS)
1739
1740  006636  104005                   ERROR   5               ;RHAS DOES NOT CLEAR BY WRITING
1741                                                            ;ALL ONES INTO IT
1742
```

```
                                        ;;*****************************************************************
                                        ;*TEST 3        TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

                                        ;*      THE NUMBER OF DRIVES PRESENT IS FOUND
                                        ;*      BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
                                        ;*      IN RHCS2 INCREMENTED FROM ZERO TO SEVEN

                                        ;*      THE BITS SET IN RHAS SHOULD GIVE DRIVES PRESENT

                                        ;*      THE DRIVE TYPE IS CHECKED TO BE AN RP04/RP06 AND THEN
                                        ;*    . UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'

                                        ;;*****************************************************************
006640  000004                  TST3:   SCOPE
006642  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
006650  012737  000003  004604          MOV     #4-1,@#TSTNM    ;THIS SAVES TEST NUMBER
006656  000005                          RESET                   ;START WITH AN INIT
006660  004737  045352                  JSR     PC,@#$TKINT     ;INITIALIZE TTY KEYBOARD

006664  032777  020000  172246          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT?
006672  001026                          BNE     4$              ;BRANCH IF YES
006674  104400  006702                  TYPE    ,65$            ;;TYPE ASCIZ STRING
006700  000423                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12><15><12>/LOOKING AT RHAS - DRIVES PRESENT /
006750                          64$:
006750  013701  002316          4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
006754  013702  002276                  MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
006760  005012                          CLR     @R2             ;CLEAR RHCS2
006762  012700  000010                  MOV     #8.,R0          ;COUNT
006766  013704  002302                  MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
006772  012714  177777          1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
006776  005212                          INC     @R2             ;INCREMENT UNIT NO.
007000  005300                          DEC     R0              ;COUNT DOWN DRIVE COUNTER
007002  001373                          BNE     1$              ;DO NEXT UNIT IF 8 NOT DONE

007004  11.137  004742                  MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
                                                                ;USED IN DRIVE CLEAR TEST
007010  105037  004743                  CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
007014  105711                          TSTB    @R1             ;TEST 'RHAS' FOR ANY DRIVES PRESENT
007016  001402                          BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
007020  000137  007372                  JMP     XE2             ;SOME THERE - GO FILL "UNITS" TABLE

007024  032777  020000  172106  2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
007032  001402                          BEQ     3$              ;"NO DRIVES" MESSAGE IF NOT
007034  000137  007730                  JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
                                                                ;"UNITS" TABLE WITH SELECTED ONE IF SO

007040                          3$:
007040  104400  007046                  TYPE    ,67$            ;;TYPE ASCIZ STRING
007044  000420                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/NO DRIVES PRESENT - RHAS = 0/
007106                          66$:    -
007106  104400  007114                  TYPE    ,69$            ;;TYPE ASCIZ STRING
```

```
1797  007112  000430                          BR      68$             ;;GET OVER THE ASCIZ
1798                                  ;;69$:  .ASCIZ  <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
1799  007174                          68$:
1800  007174  104400  007202                  TYPE    ,71$            ;;TYPE ASCIZ STRING
1801  007200  000430                          BR      70$             ;;GET OVER THE ASCIZ
1802                                  ;;71$:  .ASCIZ  <15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/
1803  007262                          70$:
1804  007262  104400  007270                  TYPE    ,73$            ;;TYPE ASCIZ STRING
1805  007266  000437                          BR      72$             ;;GET OVER THE ASCIZ
1806                                  ;;73$:  .ASCIZ  <15><12>/TO LOOP ON THIS TEST WO PRINTOUT SET SWITCHES 13, 8, 1, & 0/
1807  007366                          72$:
1808
1809  007366  000137  040714                  JMP     @#$EOP          ;GO OUT---------------------------------->
1810
1811
1812
1813                                          ;*SET UP UNITS TABLE
1814
1815  007372                          XE2:
1816  007372  012700  000010          2$:     MOV     #8.,R0          ;COUNTER
1817  007376  012703  004676                  MOV     #UNITS,R3       ;POINTER
1818  007402  012723  177777          3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
1819  007406  005300                          DEC     R0              ;COUNT
1820  007410  001374                          BNE     3$              ;BRANCH IF 8 NOT DONE
1821  007412  012703  004676                  MOV     #UNITS,R3       ;POINTER
1822  007416  005005                          CLR     R5              ;INITIALIZE UNIT NO. TO 0
1823  007420  005037  004720                  CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
1824  007424  012700  000010                  MOV     #8.,R0          ;COUNTER
1825  007430  011137  001176                  MOV     @R1,@#$TMP0     ;TEMPORARY STORAGE
1826  007434  006037  001176          4$:     ROR     @#$TMP0         ;SET CARRY IF ONE IN  C BIT
1827  007440  103120                          BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
1828
1829  007442  010577  172630                  MOV     R5,@RHCS2       ;INSERT UNIT NUMBER INTO RHCS2 UA BITS
1830  007446  022777  024020  172650          CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
1831  007454  001503                          BEQ     6$              ;TYPE THE UNIT NO. IF YES
1832  007456  022777  020020  172640          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04 ?
1833  007464  001477                          BEQ     6$              ;TYPE UNIT NO. IF YES
1834
1835                                  ;;**********************************************************************
1836  007466  022777  024021  172630          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
1837  007474  001473                          BEQ     6$              ;TYPE UNIT NO. IF SO
1838  007476  022777  020021  172620          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
1839  007504  001467                          BEQ     6$              ;TYPE UNIT NO. IF SO
1840
1841  007506  022777  024022  172610          CMP     #24022,@RHDT    ;IS THIS A DUAL PORT RP06 ?
1842  007514  001463                          BEQ     6$              ;TYPE THE UNIT NO. IF SO
1843  007516  022777  020022  172600          CMP     #20022,@RHDT    ;IS THIS A SINGLE PORT RP06 ?
1844  007524  001457                          BEQ     6$              ;TYPE UNIT NO. IF SO
1845                                  ;;**********************************************************************
1846
1847
1848                                          ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
1849                                          ;*OUT THE DEVICE TYPE
1850
```

```
1851  007526  104400  007534                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1852  007532  000410                          BR      64$             ;;GET OVER THE ASCIZ
1853                                  ;;65$:  .ASCIZ  <15><12>/UNIT NUMBER /
1854  007554                          64$:
1855  007554  010546                          MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
1856  007556  104404                          TYPDS
1857  007560  104400  007566                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1858  007564  000406                          BR      66$             ;;GET OVER THE ASCIZ
1859                                  ;;67$:  .ASCIZ  /,  RHDT = /
1860  007602                          66$:
1861  007602  017746  172516                  MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
1862  007606  104401                          TYPOC
1863  007610  104400  007616                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1864  007614  000422                          BR      68$             ;;GET OVER THE ASCIZ
1865                                  ;;69$:  .ASCIZ  ? - NOT AN RP04/RP05/RP06 DEVICE !!?
1866  007662                          68$:
1867  007662  000407                          BR      5$              ;NO RP04/RP05/RP06 FOUND SO TEST NEXT UNIT
1868
1869  007664  010523                  6$:     MOV     R5,(R3)+        ;LOAD TABLE POSITION AND INCR IT
1870  007666  104400  001223                  TYPE    ,$CRLF
1871  007672  010546                          MOV     R5,-(SP)        ;PUT DRIVE NO. ON STACK
1872  007674  104404                          TYPDS                   ;TYPE DRIVE NO.
1873  007676  005237  004720                  INC     @#NOUNIT        ;INCR THE TOTAL NO. OF UNITS
1874
1875  007702  005205                  5$:     INC     R5              ;'RHCS2' UNIT ADDRESS
1876  007704  005300                          DEC     R0              ;DRIVE COUNTER DOWN ONE
1877  007706  001252                          BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
1878
1879  007710  013737  004676  004716          MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND OR 0
1880  007716  013737  004720  004722          MOV     @#NOUNIT,@#NUNIT;SAVE NO. OF UNITS
1881  007724  005337  004722                  DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
1882                                                                  ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
1883
1884  007730  005737  004726          SELTST: TST     @#SELECT        ;STARTING ADDRESS 200
1885  007734  001403                          BEQ     TST4    ;      SKIP NEXT IF STARTING FROM 200
1886  007736  013737  004730  004716          MOV     @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE
```

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

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

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

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

                                        ;;*****************************************************************
007744  000004                    4$:   SCOPE
007746  012737  000001  001212          MOV     #1,STIMES       ;;DO 1 ITERATION
007754  012737  010706  001136          MOV     #1$,SLPADR      ;;SET SCOPE LOOP ADDRESS
007762  012706  001000                  MOV     #STACK,SP       ;RESET STACK
007766  012737  000004  004604          MOV     #4,@#TSTNM      ;SAVE TEST NUMBER

007774  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
010000  005037  004740                  CLR     @#ATTENT        ;CLEAR

010004  005737  004716                  TST     @#UNIT          ;IS "UNIT 0" NEXT IN THE UNITS TABLE ?
010010  001107                          BNE     9$              ;IF NOT TEST THIS UNIT
010012  012700  000041                  MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
010016  122710  000011                  CMPB    #11,(R0)        ;WAS IT AN RP04/5/6 ?
010022  001102                          BNE     9$              ;NO - GO AHEAD WITH TESTING UNIT #0

010024  005737  004726                  TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
                                                                ;(IE. WAS IT A 210 START ?)
010030  001006                          BNE     12$             ;IF SO, CHANGE PACKS

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

010032  012700  004676                  MOV     #UNITS,R0       ;LOAD UNITS TABLE POINTER
010036  005720                          TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
                                                                ;(DOUBLE INCREMENT THE POINTER, R0)
010040  022710  177777                  CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
010044  001065                          BNE     11$             ;IF SO, USE THE NEXT DRIVE & DEC "NOUNITS"
                                                                ;IF NOT, MUST USE DRIVE #0 & CHANGE PACK
010046                          12$:
010046  104400  010054                  TYPE    ,65$            ;;TYPE ASCIZ STRING
010052  000434                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12><15><12>/DISMOUNT PACK FROM UNIT #0 AND MOUNT A SCRATCH PACK
010144                          64$:
010144  104400  010152                  TYPE    ,67$            ;;TYPE ASCIZ STRING
010150  000421                          BR      66$             ;;GET OVER THE ASCIZ
```

```
                                        ;;67$:  .ASCIZ  <15><12> PRESS CONTINUE WHEN FINISHED/<15><12>
                                        66$:
010214                                          HALT
010214  000000                                    BR      9$              ;CONTINUE, USING SCRATCH PACK ON UNIT #0
010216  000434
                                        11$:    MOV     (R0),@#UNIT     ;SET UP TO BE THE UNIT UNDER TEST
010220  011037  004716                          DEC     @#NOUNITS       ;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
010224  005337  004720
                                        9$:     MOV     @#UNIT,R0       ;R0 CONTAINS UNIT UNDER TEST
010230  013700  004716


                                        ;;*****************************************************************
010234  005037  004744                          CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
010240  010077  172032                          MOV     R0,@RHCS2       ;SET UP UNIT ADDRESSING
010244  022777  024022  172052                  CMP     #24022,@RHDT    ;IS IT A DUAL PORT RP06 ?
010252  001405                                  BEQ     2$              ;YES...SET THE FLAG
010254  022777  020022  172042                  CMP     #20022,@RHDT    ;IS IT A SINGLE PORT RP06 ?
010262  001401                                  BEQ     2$              ;YES...SET FLAG
010264  000404                                  BR      3$              ;DON'T SET FLAG - CHECK FOR RP04
010266  012737  177777  004744          2$:     MOV     #-1,@#RP06      ;SET THE FLAG
010274  000416                                  BR      8$              ;DON'T CHECK FOR RP04, IT WAS RP06

010276  005037  004746                  3$:     CLR     @#RP05          ;CLEAR MEMOREX RP04 DEVICE FLAG
010302  022777  024021  172014                  CMP     #24021,@RHDT    ;IS IT A DUAL PORT MEMOREX RP04 ?
010310  001405                                  BEQ     7$              ;YES..SET THE FLAG FOR ADDR PLUG TESTS
010312  022777  020021  172004                  CMP     #20021,@RHDT    ;IS IT A SINGLE PORT MEMOREX RP04 ?
010320  001401                                  BEQ     7$              ;YES..SET THE FLAG FOR ADDR PLUG TESTS
010322  000403                                  BR      8$              ;DON'T SET FLAG - NOT MEMOREX DRIVE
010324  012737  177777  004746          7$:     MOV     #-1,@#RP05      ;SET THE FLAG
010332                                  8$:                             ;ASSUME THE NEXT UNIT IS AN RP04
                                        ;;*****************************************************************


010332  116037  004666  004740                  MOVB    ATABLE(R0),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
010340  104400  010346                          TYPE    ,69$            ;;TYPE ASCIZ STRING
010344  000414                                  BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12>/TESTING DRIVE NUMBER/
010376                                  68$:
010376  013746  004716                          MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
010402  104404                                  TYPOS                   ;TYPE  DRIVE NO.
010404  104400  010412                          TYPE    ,71$            ;;TYPE ASCIZ STRING
010410  000410                                  BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  <15><12> SERIAL NO. = /
010432                                  70$:
010432  017746  171670                          MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
010436  104401                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
010440  104400  010446                          TYPE    ,73$            ;;TYPE ASCIZ STRING
010444  000410                                  BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  <15><12>/DRIVE TYPE = /
010466                                  72$:
010466  017746  171632                          MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
010472  104401                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

1995

```
                                        ;TYPE OUT THE DRIVE TYPE IN ASCII
                                        ;
                                        ;
                                        ;
                                ;;**************************************************************
010474  022777  024020  171622          CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
010502  001425                          BEQ     4$              ;TYPE ASCII MSG OUT
010504  022777  020020  171612          CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
010512  001421                          BEQ     4$              ;TYPE THE MESSAGE
                                        ;
010514  022777  024021  171602          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
010522  001453                          BEQ     6$              ;TYPE THE MESSAGE
010524  022777  020021  171572          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
010532  001447                          BEQ     6$              ;TYPE THE MESSAGE
                                        ;
010534  022777  024022  171562          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
010542  001424                          BEQ     5$              ;TYPE THE MESSAGE
010544  022777  020022  171552          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
010552  001420                          BEQ     5$              ;TYPE THE MESSAGE
010554  000454                          BR      1$              ;DRIVE IS NOT AN RP04/RP05/RP06
                                                                ;DON'T TYPE ASCII MESSAGE OUT
                                        ;
                                                                ;-SHOULD NEVER HAPPEN AT THIS POINT
                                                                ;UNLESS DRIVE GOT SICK WHILE TESTING
                                                                ;WAS IN PROGRESS
                                        ;
010556                          4$:
010556  104400  010564                  TYPE    ,75$            ;;TYPE ASCIZ STRING
010562  000413                          BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  <15><12>/DRIVE IS AN RP04/<15><12>
010612                          74$:
010612  000435                          BR      1$              ;SKIP NEXT MESSAGE
010614                          5$:
010614  104400  010622                  TYPE    ,77$            ;;TYPE ASCIZ STRING
010620  000413                          BR      76$             ;;GET OVER THE ASCIZ
                                ;;77$:  .ASCIZ  <15><12>/DRIVE IS AN RP06/<15><12>
010650                          76$:
010650  000416                          BR      1$              ;SKIP NEXT MESSAGE
010652                          6$:
010652  104400  010660                  TYPE    ,79$            ;;TYPE ASCIZ STRING
010656  000413                          BR      78$             ;;GET OVER THE ASCIZ
                                ;;79$:  .ASCIZ  <15><12>/DRIVE IS AN RP05/<15><12>
010706                          78$:
                                ;;**************************************************************
                                        ;
                                        ;
                                        ;
                                        ;
010706  005777  171414          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
010712  005777  171406                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
010716  017737  171404  002410          MOV     @RHSN,@#SN      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
010724  017737  171374  002406          MOV     @RHDT,@#DT      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 5         CHECK MOL TO BE HIGH
                                        ;*
                                        ;*      MAKE SURE THAT DRIVE IS ON LINE BEFORE STARTING PROGRAM
                                        ;*      IF DRIVE IS OFFLINE, THEN AFTER TYPE OUT THE PROGRAM WILL
                                        ;*      HANG FOR EVER WAITING FOR DRIVE TO GO ON LINE.
                                        ;;*****************************************************************
010732  000004                  TST5:   SCOPE
010734  012737  000005  004604          MOV     #6-1,@#TSTNM    ;THIS SAVES TEST NUMBER
010742  004737  041366                  JSR     PC,@#CLDISK     ;GIVE INITIALIZE
010746  032713  010000                  BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
010752  001151                          BNE     TST6    :       BRANCH IF MOL HIGH
010754  104400  010762                  TYPE    ,65$            ;;TYPE ASCIZ STRING
010760  000420                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/DRIVE IS OFFLINE - MOL IS LOW/
011022                                  64$:
011022  104400  011030                  TYPE    ,67$            ;;TYPE ASCIZ STRING
011026  000424                          BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/HIT START ON DRIVE TO GET IT ON LINE/
011100                                  66$:
011100  104400  011106                  TYPE    ,69$            ;;TYPE ASCIZ STRING
011104  000431                          BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS HIGH
011170                                  68$:
011170  032713  010000                  1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
011174  001775                          BEQ     1$              ;BRANCH IF MOL IS HIGH
011176  104400  011204                  TYPE    ,71$            ;;TYPE ASCIZ STRING
011202  000435                          BR      70$             ;;GET OVER THE ASCIZ
011270                                  ;;71$:  .ASCIZ  <15><12>/GOOD - MOL IS NOW HIGH. PROGRAM WILL NOW BE EXECUTED <15><12>
011270                                  70$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 6         PROGRAM INTERRUPT
                                        ;*
                                        ;*      PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
                                        ;*      IN RHCS1 AT THE SAME TIME
                                        ;*      THIS SHOULD INTERRUPT THROUGH LOCATION 254
                                        ;*      THE PROCESSOR PRIORITY IS SET TO 4
                                        ;;*****************************************************************
011276  000004                    TST6:   SCOPE

011300  012737  000006  004604          MOV     #7-1,@#TSTNM    ;THIS SAVES TEST NUMBER
011306  012706  001000                  MOV     #STACK,SP       ;RESET STACK

011312  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

011316  013700  002266                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
011322  012720  011370                  MOV     #RPTRP1,(R0)+   ;THIS IS FOR TIMELY INTERRUPTS
011326  012710  000340                  MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
                                                                ;PRIORITY = 7
011332  012737  000200  177776          MOV     #200,PS         ;SET PROCESSOR PRIORITY @ 4 (DISK @ 5)
011340  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT

011344  013737  042064  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
011352  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
011356  001375                  BNE     1$                      ;BRANCH IF NOT ZERO
                                                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
011360  104065                  ERROR   65                      ;INTERRUPT DID NOT OCCUR
011362  012712  000040          MOV     #40,@R2                 ;CLEAR CONTROLLER VIA CS2 CLR BIT
011366  000407                  BR      TST7    ;              BRANCH TO NEXT TEST ----------------

011370  022626                  RPTRP1: CMP     (SP)+,(SP)+     ;RESTORE STACK
011372  022711  004200                  CMP     #DVA!RDY,@R1    ;IE SHOULD BE LOW AS RUPT OCCURRED
011376  001403                  BEQ     TST7    ;              CONTINUE IF GOOD ----------------
011400  104065                  ERROR   65                      ;INTERRUPT OCCURED BUT
                                                                ;IE FAILED TO RESET
011402  012712  000040          MOV     #40,@R2                 ;CLEAR CONTROLLER VIA CS2 CLR BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 7         INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
                                ;
                                ;*      PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
                                ;*      IE AND RDY IS SET. THIS SHOULD NOT ALLOW INTERRUPT
                                ;;*****************************************************************
011406  000004                  TST7:   SCOPE
011410  012737  000007  004604          MOV     #10-1,@#TSTNM   ;THIS SAVES TEST NUMBER
011416  012706  001000          MOV     #STACK,SP       ;RESET STACK
011422  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
011426  013700  002266          MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
011432  012720  011476          MOV     #RPTRP2,(R0)+   ;THIS IS FOR UNTIMELY INTERRUPTS
011436  012710  000340          MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
                                                        ;PRIORITY = 7
011442  012737  000240  177776          MOV     #240,PS         ;SET PROCESSOR PRIORITY @ 5
011450  012711  000300          MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT
011454  013737  042064  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
011462  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
011466  001375                  BNE     1$              ;BRANCH IF NOT ZERO
                                                        ;BEFORE THIS IS ZERO INTERRUPT WHOLLD
                                                        ;OCCUR
011470  012712  000040          MOV     #40,@R2         ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
011474  000404                  BR      TST10   ;       NO INTERRUPT SO CONTINUE ----------------)
011476  022626                  RPTRP2: CMP     (SP)+,(SP)+     ;RESTORE STACK
011500  104065                  ERROR   65              ;INTERRUPT OCCURRED WITH PROCESSOR
                                                        ;PROCESSOR STATUS SAME AS DISK
011502  012712  000040          MOV     #40,@R2         ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 10       PACK ACKNOWLEDGE
                                ;*
                                ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
                                ;*
                                ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                ;*      THEN THIS TEST IS NOT PERFORMED
                                ;*
                                ;*      IF NO ACT-11 MONITOR IS PRESENT
                                ;*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
                                ;*
                                ;*      THIS TESTS THE ACKNOWLEDGE COMMAND=44
                                ;*      VV BIT - RHDS1 BIT #6
                                ;*      MOL BIT - RHDS1 BIT #12
                                ;*      DVA BIT - RHCS1 BIT #11
                                ;*      THE DRIVE IS STOPED MOL IS CHECKED TO BE 0
                                ;*      AND DVA SHOULD BE 0
                                ;*      THE DRIVE IS POWERED UP
                                ;*      VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1
                                ;*      GO SHOULD BE 0
                                ;*      ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED
                                ;*      PACK ACKNOWLEDGE IS ISSUED
                                ;*      ALL STORED REGISTERS SHOULD BE UNCHANGED
                                ;*      EXCEPT VV
                                ;*****************************************************************
011506  000004                  TST10:  SCOPE
011510  012706  001000                  MOV     #STACK,SP       ;RESET STACK
011514  012737  000010  004604          MOV     #10.,@#TSTNM    ;SAVE TEST NUMBER

011522  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
011526  012737  000000  177776          MOV     #0,PS           ;SET PROCESSER STATUS TO 0

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

011534  005737  004724                  TST     @#NOPUSH        ;IS THIS A 220 START ?
011540  001007                          BNE     1$              ;SKIP THIS TEST IF SO
011542  005737  000042                  TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
011546  001004                          BNE     1$              ;SKIP THIS TEST
011550  005737  001100                  TST     @#$PASS         ;FIRST PASS ?
011554  001001                          BNE     1$              ;SKIP THIS TEST IF NOT
011556  000402                          BR      2$              ;CONTINUE WITH THIS TEST

011560                          1$:
011560  000137  012266                  JMP     TST11   ;       JUMP TO NEXT TEST ----------------)
011564                          2$:
011564  104400  011572                  TYPE    ,65$            ;;TYPE ASCIZ STRING
011570  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/STOP DRIVE /
```

```
2218  011610                          64$:
2219  011610  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
2220  011614  104404                          TYPDS
2221  011616  104400  011624                  TYPE    ,67$            ;;TYPE ASCIZ STRING
2222  011622  000413                          BR      66$             ;;GET OVER THE ASCIZ
2223                                  ;;67$:  .ASCIZ  / THEN HIT CONTINUE/<15><12>
2224  011652                          66$:
2225  011652  000000                          HALT
2226  011654  032713  010000                  BIT     #MOL,@R3        ;MOL IN RHDS1 SHOULD BE = 0
2227  011660  001403                          BEQ     3$              ;BRANCH IF MOL=0
2228  011662  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER ADDRESS-RHDS1
2229  011666  104010                          ERROR   10              ;ON SPINPLE POWERED DOWN
2230                                                                  ;MOL SHOULD BE 0
2231  011670  013746  004716          3$:     MOV     @#UNIT,-(SP)    ;UNIT NUMBER
2232  011674  052716  000100                  BIS     #IR,(SP)        ;INCLUDE IR
2233  011700  022612                          CMP     (SP)+,@R2       ;ONLY UNIT NO. AND IR SHOJLD BE
2234                                                                  ;HIGH IN RHCS2
2235  011702  001403                          BEQ     4$              ;BRANCH IF RHCS2 GOOD
2236  011704  010237  001122                  MOV     R2,@#$BDADR     ;FAILING REGISTER ADDRESS-RHCS2
2237  011710  104011                          ERROR   11              ;WITH SPINDLE POWERED DOWN
2238                                                                  ;ONLY UNIT NO. AND IR SHOULD BE
2239                                                                  ;HIGH
2240
2241  011712                          4$:
2242
2243  011712  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2244                                                                  ;R3-RHDS1, R4-RHER1
2245                                                                  ;GIVE RH-11 INITIALIZE
2246                                                                  ;SETUP UNIT NUMBER
2247  011716  104400  011724                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2248  011722  000407                          BR      68$             ;;GET OVER THE ASCIZ
2249                                  ;;69$:  .ASCIZ  <15><12>/START DRIVE/
2250  011742                          68$:
2251  011742  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
2252  011746  104404                          TYPDS
2253  011750  104400  011756                  TYPE    ,71$            ;;TYPE ASCIZ STRING
2254  011754  000420                          BR      70$             ;;GET OVER THE ASCIZ
2255                                  ;;71$:  .ASCIZ  / AFTER HEAD LOAD HIT CONTINUE/<15><12>
2256  012016                          70$:
2257  012016  000000                          HALT
2258  012020  032713  010000                  BIT     #MOL,@R3        ;MOL IN RHDS1 SHOULD BE = 1
2259  012024  001411                          BEQ     5$              ;BRANCH IF MOL = 0
2260  012026  032713  000400                  BIT     #DPR,@R3        ;DPR IN RHDS1 SHOULD BE = 1
2261  012032  001406                          BEQ     5$              ;BRANCH IF DPR = 0
2262  012034  032713  000200                  BIT     #DRY,@R3        ;DRY IN RHDS1 SHOULD BE = 1
2263  012040  001403                          BEQ     5$              ;BRANCH IF DRY =0
2264  012042  032713  000100                  BIT     #VV,@R3         ;VV IN RHDS1 SHOULD BE = 0
2265  012046  001403                          BEQ     6$              ;BRANCH IF VV = 0 (GOOD)
2266  012050  010337  001122          5$:     MOV     R3,@#$BDADR     ;FAILING REGISTER ADDRESS - RHDS1
2267  012054  104012                          ERROR   12              ;WITH SPINDLE POWERED UP
2268                                                                  ;RHDS1 SHOULD HAVE VV = 0, MOL = 1
2269
2270  012056  011100                  6$:     MOV     @R1,R0          ;GET RHCS1 CONTENTS
2271  012060  042700  160076                  BIC     #SC!TRE!MCPE!76,R0 ;CLEAR SC,TRE,MCPE AND
```

```
2272
2273  012064  022700  004200                CMP     #DVA!RDY,R0     ;ALL FUNCTION BITS
2274                                                                  ;RHCS1 SHOULD HAVE
                                                                      ;GO = 0, DVA = 1, RDY = 1
2275  012070  001403                        BEQ     7$              ;BRANCH IF RHCS1 IS GOOD
2276  012072  010137  001122                MOV     R1,@#$BDADR     ;FAILING REGISTER RHCS1
2277  012076  104013                        ERROR   13              ;AFTER A POWER UP WITHOUT ANY
2278                                                                  ;INIT RHCS1 SHOULD HAVE
2279                                                                  ;GO = 0, DVA = 1, RDY = 1, IE = 0
2280                                                                  ;DISREGARD ALL OTHER BITS
2281
2282  012100  005777  170230        7$:     TST     @RHCC           ;TEST RHCC. IT SHOULD = 0
2283  012104  001411                        BEQ     10$             ;BRANCH IF RHCC = 0
2284  012106  013737  002334  004600        MOV     @#RHCC,@#REGADR ;FAILING REGISTER RHCC
2285  012114  005037  001124                CLR     @#$GDDAT        ;RHCC SHOULD BE = 0
2286  012120  017737  170210  001126        MOV     @RHCC,@#$BDDAT  ;BAD DATA
2287  012126  104014                        ERROR   14              ;AFTER POWER UP RHCC
2288                                                                  ;SHOULD BE 0
2289
2290  012130                        10$:
2291
2292  012130  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2293                                                                  ;R3-RHDS1, R4-RHER1
2294                                                                  ;GIVE RH-11 INITIALIZE
2295                                                                  ;SETUP UNIT NUMBER
2296  012134  004737  041424                JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
2297                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
2298  012140  104400  066402                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
2299  012144  000000                        HALT                    ;STOP
2300  012146  013777  002460  170124        MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
2301                                                                  ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
2302
2303
2304                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2305
2306  012154  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
2307  012160  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2308  012162  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
2309                                                                  ;THE REGISTERS ARE SAVED
2310  012164  000022                        18.                     ;NUMBER OF REGISTERS
2311                                                                  ;SAVED = 18.
2312
2313  012166  013777  004606  170072        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2314                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
2315                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2316                                                                  ;'TIME' WILL ONLY SAVE
2317                                                                  ;CURRENT CYLINDER ADDRESS
2318                                                                  ;AND LOOK AHEAD REGISTERS
2319
2320
2321  012174  013746  002460                MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
2322  012200  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
2323                                                                  ;WITHOUT INTERRUPT ENABLE
2324  012204  012677  170070                MOV     (SP)+,@RHCS1    ;GO WITH
2325                                                                   ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
```

```
                                                        ;WITH INTERRUPT DISABLED


012210  104412                          WAT             ;WAIT FOR VV BIT TO SET
012212  002322                          RHDS1           ;WAIT FOR RHDS1 REGISTER
012214  000100                          VV              ;WAIT FOR VV BIT IN RHDS1 REGISTER
012216  000001                          1.              ;ALLOW 10 MICRO SECONDS
012220  000001                          1.              ;VV MUST SET BETWEEN
                                                        ;00 AND 20 MICRO SECONDS


012222  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
012226  002322                          RHDS1           ;CHANGE RHDS1 REGISTER

012230  000001                          1               ;1 BIT/BITS TO BE CHANGED
012232  000001                          1               ;NEW VALUE OF VV IS 1
012234  000100                          VV              ;CHANGE VV BIT

                                        ;*NOW COMPARE REGISTERS SO THAT NO REGISTERS
                                        ;*CHANGED EXCEPT VV IN RHDS1 AND IE IN RHCS1

012236  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
012242  004612                          SAVERE          ;GOOD DATA SAVED IN 'SAVERE'
012244  002354                          WC              ;TEST DATA STARTING FROM 'RHWC'
012246  000022                          18.             ;18. REGISTERS TO BE COMPARED
012250  012254                          11$             ;RETURN TO 11$ ON ERROR
012252  012260                          12$             ;RETURN TO 12$ ON NO ERROR

012254  104015                  11$:    ERROR   15      ;GIVING A PACK ACKNOWLEDGE
012256  000207                          RTS     PC      ;CAUSED AN ERROR
                                                        ;PACK ACKNOWLEDGE SHOULD
                                                        ;SET VV IN RHDS1
                                                        ;INTERRUPT SHOULD MAKE
                                                        ;IE = 0
                                                        ;NO OTHER REGISTERS SHOULD
                                                        ;CHANGE
                                                        ;GOOD DATA GIVES
                                                        ;CONTENTS OF REGISTER BEFORE
                                                        ;PACK ACKNOWLEDGE
                                                        ;RECEIVED DATA GIVES
                                                        ;CONTENTS OF REGISTER
                                                        ;AFTER PACK ACKNOWLEDGE

012260  012737  177777  046760  12$:    MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
```

```
2374
2375
2376
2377                                    ;;*****************************************************************
2378                                    ;*TEST 11       SET VV BIT #6 IN RHDS1
2379                                    ;*      THIS TEST SETS VV IN RHDS1 INCASE
2380                                    ;*      ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
2381                                    ;*      IS NOT PERFORMED
2382                                    ;*      THERE IS A RESET AT THE BEGINING OF THIS TEST
2383                                    ;*      FOR ERROR RECOVERY ONLY.
2384
2385                                    ;;*****************************************************************
2386  012266  000004                    TST11:  SCOPE
2387
2388                                            ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
2389                                            ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
2390  012270  000005                    RESET
2391  012272  004737  045352                   JSR     PC,@#$TKINT     ;INITILIZE TK
2392
2393
2394  012276  012706  001000                   MOV     #STACK,SP       ;RESET STACK
2395  012302  012737  000011  004604           MOV     #11,@#TSTNM     ;SAVE TEST NUMBER
2396
2397  012310  004737  041366                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2398                                                                    ;R3-RHDS1, R4-RHER1
2399                                                                    ;GIVE RH-11 INITIALIZE
2400                                                                    ;SETUP UNIT NUMBER
2401  012314  004737  041424                   JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
2402                                                                    ;AND THAT NO STATUS BITS ARE STUCK = 1
2403  012320  104400  066402                   TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
2404  012324  000000                           HALT                    ;STOP
2405  012326  013777  002460  167744           MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
2406                                                                    ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
2407
2408
2409                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2410
2411  012334  004037  041534                   JSR     R0,@#SAVER      ;SAVE REGISTERS
2412  012340  002272                           RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2413  012342  004612                           SAVERE                  ;STARTING ADDRESS OF WHERE
2414                                                                    ;THE REGISTERS ARE SAVED
2415  012344  000022                           18.                     ;NUMBER OF REGISTERS
2416                                                                    ;SAVED = 18.
2417
2418  012346  013777  004606  167712           MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2419                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
2420                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
2421                                                                    ;'TIME' WILL ONLY SAVE
2422                                                                    ;CURRENT CYLINDER ADDRESS
2423                                                                    ;AND LOOK AHEAD REGISTERS
2424
2425
2426  012354  013746  002460                   MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
2427  012360  052716  000001                   BIS     #GO,(SP)        ;GET READY TO SET GO
```

```
2428
2429  012364  012677  167710                    MOV     (SP)+,@RHCS1    ;WITHOUT INTERRUPT ENABLE
2430                                                                      ;GO WITH
2431                                                                      ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
2432                                                                      ;WITH INTERRUPT DISABLED
2433  012370  011100                            MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
2434  012372  011305                            MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
2435
2436
2437  012374  104412                            WAT                     ;WAIT FOR VV BIT TO SET
2438  012376  002322                            RHDS1                   ;WAIT FOR RHDS1 REGISTER
2439  012400  000100                            VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
2440  012402  000001                            1.                      ;ALLOW 10 MICRO SECONDS
2441  012404  000001                            1.                      ;VV MUST SET BETWEEN
2442                                                                      ;00 AND 20 MICRO SECONDS
2443
2444                                            ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
2445                                            ;*R0 AND R5 IMMEDIATELY AFTER GO
2446
2447  012406  013746  002460                    MOV     @#PKACK,-(SP)   ;SAVE COMMAND
2448  012412  052716  004200                    BIS     #DVA!RDY,(SP)   ;INCLUDE DVA!RDY
2449  012416  011637  001124                    MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
2450  012422  022600                            CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!RDY
2451                                                                      ;AND COMMAND SHOULD BE SET
2452  012424  001405                            BEQ     64$             ;BRANCH IF GOOD
2453  012426  010037  001126                    MOV     R0,@#$BDDAT     ;BAD DATA
2454  012432  010137  004600                    MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
2455  012436  104021                            ERROR   21              ;DURING ABOVE OPERATION ONLY
2456                                                                      ;COMMAND AND DVA!RDY SHOULD BE SET
2457  012440  012746  010700          64$:      MOV     #MOL!DPR!DRY!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
2458  012444  011637  001124                    MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
2459  012450  022605                            CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!DRY!VV
2460                                                                      ;SHOULD BE SET
2461  012452  001405                            BEQ     66$             ;BRANCH IF GOOD
2462  012454  010537  001126                    MOV     R5,@#$BDDAT     ;BAD DATA
2463  012460  010337  004600                    MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
2464  012464  104063                            ERROR   63              ;DURING ABOVE OPERATION ONLY
2465                                                                      ;MOL!DPR!DRY!VV SHOULD BE SET
2466  012466                          66$:
2467
2468  012466  004037  042246                    JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2469  012472  002322                            RHDS1                   ;CHANGE RHDS1 REGISTER
2470
2471  012474  000001                            1                       ;1 BIT/BITS TO BE CHANGED
2472  012476  000001                            1                       ;NEW VALUE OF VV IS 1
2473  012500  000100                            VV                      ;CHANGE VV BIT
2474
2475                                            ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
2476                                            ;*WITH AFTER PACK ACKNOWLEDGE
2477
2478
2479
2480  012502  004037  042354                    JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2481                                                                      ;PRESENT VALUE
```

```
012506  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
012510  002354                          #C                      ;TEST DATA STARTING FROM 'RHWC'
012512  000022                          18.                     ;18. REGISTERS TO BE COMPARED
012514  012520                          1$                      ;RETURN TO 1$ ON ERROR
012516  012524                          2$                      ;RETURN TO 2$ ON NO ERROR

012520  104215                  1$:     ERROR   15              ;GIVING A PACK ACKNOWLEDGE
012522  000207                          R*S     PC              ;CAUSED AN ERROR
                                                                ;PACK ACKNOWLEDGE SHOULD
                                                                ;SET VV IN RHDS1
                                                                ;INTERRUPT SHOULD MAKE
                                                                ;IE = 0
                                                                ;NO OTHER REGISTERS SHOULD
                                                                ;CHANGE
                                                                ;GOOD DATA GIVES CONTENTS
                                                                ;OF REGISTER BEFORE COMMAND
                                                                ;RECEIVED DATA GIVES CONTENTS
                                                                ;OF REGISTER AFTER COMMAND
012524                          2$:
```

```
                                ;;*****************************************************************
012524  005737  004744          ;       TST     @#RPO6          ;TEST FOR RPO6 DRIVE
012530  001005                  ;       BNE     3$              ;IF = 1 DO "MAKECYL" 777
012532  005737  004746          ;       TST     @#RPO5          ;TEST FOR RPO5 DRIVE
012536  001004                  ;       BNE     4$              ;IF = 1 DO "MAKECYL" 377
012540  000137  014664          ;       JMP     TST16   ;       IF BOTH = 0, DON'T DO EITHER "MAKECYL" ------,
                                ;                               ;OR THE ADDRESS PLUG TESTS
012544                          ;3$:
012544  000137  012610          ;       JMP     TST13   ;       DO "MAKECYL" 777 ---------------------------,
012550                          ;4$:
                                ;;*****************************************************************



                                ;;*****************************************************************
                                ;*TEST 12        MAKE CURRENT CYLINDER = 377
                                ;;*****************************************************************
012550  000004                  ;TST12: SCOPE
012552  012706  001000          ;       MOV     #STACK,SP       ;RESET STACK
012556  012737  000012  004604  ;       MOV     #13-1,@#TSTNM   ;THIS SAVES TEST NUMBER

012564  004737  041366          ;       JSR     PC,@#CLDISK     ;INIT DRIVE
012570  012777  000001  167522  ;       MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
012576  004037  041036          ;       JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                ;                               ;COMMAND FOLLOWED BY A INIT
                                ;                               ;THIS SHOULD CHANGE RHCS TO 377
012602  000377                  ;       377

012604  000137  012644          ;       JMP     TST14   ;       SKIP NEXT "MAKECYL" ------------------------)



                                ;;*****************************************************************
                                ;*TEST 13        MAKE CURRENT CYLINDER = 777
                                ;;*****************************************************************
012610  000004                  ;TST13: SCOPE
012612  012706  001000          ;       MOV     #STACK,SP       ;RESET STACK
012616  012737  000013  004604  ;       MOV     #14-1,@#TSTNM   ;THIS SAVES TEST NUMBER

012624  004737  041366          ;       JSR     PC,@#CLDISK     ;INIT DRIVE
012630  012777  000001  167462  ;       MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
012636  004037  041036          ;       JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                ;                               ;COMMAND FOLLOWED BY A INIT
                                ;                               ;THIS SHOULD CHANGE RHCS TO 777
012642  000777                  ;       777
```

```
                                        ;;*****************************************************************
                                        ;*TEST 14       ADDRESS PLUG CHANGE ERROR

                                        ;*      CHECK PROPER ADDRESS PLUG FUNCTIONALITY
                                        ;*      BY PULLING THE ADDRESS PLUG DURING A COMMAND ISSUED
                                        ;*      IN DIAGNOSTIC MODE (TO GUARANTEE COMMAND IS STILL ACTIVE WHEN
                                        ;*      PLUG IS PULLED) AND VERIFYING THAT THE DRIVE GOES OFF LINE

                                        ;*      THE ADDRESS PLUG IS THEN REPLACED AND RETURN OF THE PROPER
                                        ;*      RESPONDING DRIVE IS VERIFIED, AS WELL AS THE FACT THAT
                                        ;*      ATTENTION BITS COME UP IN THE PROPER BIT LOCATION(?????), AND THE
                                        ;*      ADDRESS CHANGE ERROR (ACE) - RHER3 BIT #13 IS SET, AS WELL AS
                                        ;*      SC,TRE,ERR,AND MCPE; VV IS RESET AND RHCC = 000 (DRIVE RECALIBRATED)

                                        ;*      IN ADDITION VERIFICATION IS ALSO MADE THAT NO OTHER DRIVE
                                        ;*      NUMBERS APPEAR ON THE BUSS AFTER THE PLUG IS REPLACED. (??????)

                                        ;;*****************************************************************
012644  000004                  TST14:  SCOPE
012646  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION

                                        ;;*****************************************************************
012654  005737  004744                  TST     @#RP06          ;TEST FOR RP06 DRIVE
012660  001005                          BNE     4$              ;IF = 1, DO THIS TEST
012662  005737  004746                  TST     @#RP05          ;TEST FOR MEMOREX RP04
012666  001002                          BNE     4$              ;IF = 1, DO THIS TEST
                                                                ;IF NEITHER FLAG IS UP, ASSUME THE
                                                                ;DRIVE IS AN ISS RP04 AND SKIP TEST
012670  000137  013656                  JMP     TST15   ;       JUMP TO NEXT TEST ----------------)
012674                          4$:
                                        ;;*****************************************************************

012674  012706  001000                  MOV     #STACK,SP       ;RESET STACK
012700  012737  000014  004604          MOV     #14,@#STNM      ;SAVE TEST NUMBER

012706  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
012712  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
012716  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
012722  000000                          HALT                    ;STOP

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

012724  005737  004724                  TST     @#NOPUSH        ;IS THIS A 220 START ?
012730  001007                          BNE     1$              ;SKIP THIS TEST IF SO
012732  005737  000042                  TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
012736  001004                          BNE     1$              ;SKIP THIS TEST
012740  005737  001100                  TST     @#$PASS         ;FIRST PASS ?
```

```
002604  012744  001001                        BNE     1$              ;SKIP THIS TEST IF NOT
002605  012746  000402                        BR      2$              ;CONTINUE WITH THIS TEST
002606
002607  012750                          1$:
002608  012750  000137  013656                JMP     TST15   ;       JUMP TO NEXT TEST ----------,
002609  012754                          2$:
002610
002611                                          ;*SET DIAGNOSTIC MODE TO ENABLE A COMMAND ACTIVE WHILE
002612                                          ;*THE PLUG IS PULLED
002613
002614  012754  052777  000001  167336          BIS     #DMD,@RHMR      ;SET UP DIAGNOSTIC MODE
002615
002616                                          ;*TAKE AN INITIAL REGISTER SNAPSHOT
002617
002618  012762  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
002619  012766  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
002620  012770  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
002621                                                                  ;THE REGISTERS ARE SAVED
002622  012772  000022                          18.                     ;NUMBER OF REGISTERS
002623                                                                  ;SAVED = 18.
002624
002625
002626                                          ;*ISSUE A COMMAND AND THE 'GO' BIT (NOT POSITIONING COMMAND)
002627                                          ;*TO VERIFY COMMAND ABORT IF PLUG IS PULLED
002628
002629
002630
002631                                          ;*ISSUE SOME CLOCKS TO GET THE COMMAND STARTED
002632                                          ;*(USE "SEARCH" WITH "DTETST" FLAG UP TO STOP CLOCKING ?)
002633
002634
002635
002636  012774  104400  013002                  TYPE    ,65$            ;;TYPE ASCIZ STRING
002637  013000  000423                          BR      64$             ;;GET OVER THE ASCIZ
002638                                  ;;65$:  .ASCIZ  <15><12>/REMOVE  ADDRESS PLUG ON DRIVE/
002639  013042                          64$:
002640  013042  013746  004716                  MOV     @#UNIT,-(SP)    ;GET THE UNIT NO. UNDER TEST
002641  013046  104404                          TYPDS                   ;TYPE IT OUT
002642  013050  104400  013056                  TYPE    ,67$            ;;TYPE ASCIZ STRING
002643  013054  000415                          BR      66$             ;;GET OVER THE ASCIZ
002644                                  ;;67$:  .ASCIZ  /       THEN PRESS CONTINUE/
002645  013110                          66$:
002646  013110  000000                          HALT                    ;WAIT FOR OPERATOR PLUG CHANGE
002647
002648                                          ;*CHECK THAT THE UNIT NO. UNDER TEST HAS GONE OFFLINE
002649
002650  013112  017700  167166                  MOV     @RHDST,R0       ;ATTEMPT TO ADDRESS THE DRIVE
002651  013116  004737  043300                  JSR     PC,@#PLTREG     ;TAKE REGISTER SNAPSHOTS
002652  013122  032737  010000  002360          BIT     #NED,@#CS2      ;TEST FOR NON EXISTENT DRIVE
002653  013130  001001                          BNE     7$              ;CONTINUE IF 'NED' BIT SET (UNIT
002654                                                                  ;IS OFFLINE AS IT SHOULD BE)
002655  013132  104101                          ERROR   101             ;UNIT DID NOT GO OFFLINE WHEN ADDRESS
002656                                                                  ;PLUG WAS REMOVED
002657  013134                          7$:
```

```
                                        ;*NOW REPLACE THE ADDRESS PLUG
                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
013134  104400  013142                BR      68$             ;;GET OVER THE ASCIZ
013140  000425
                              ;;69$:    .ASCIZ  <15><12>/REPLACE ADDRESS PLUG ON DRIVE/
                              68$:
013202
013202  013746  004716                MOV     @#UNIT,-(SP)    ;GET THE UNIT UNDER TEST
013206  104404                        TYPDS                   ;TYPE IT OUT
013210  104400  013216                TYPE    ,71$            ;;TYPE ASCIZ STRING
013214  000416                        BR      70$             ;;GET OVER THE ASCIZ
                              ;;71$:    .ASCIZ  /       THEN PRESS CONTINUE/<15><12>
                              70$:
013252
013252  000000                        HALT                    ;WAIT FOR OPERATOR PLUG REPLACEMENT

                                        ;*CHECK THAT THE ORIGINAL UNIT HAS COME BACK ON LINE

013254  004737  043300                JSR     PC,@#PJTREG     ;TAKE NEW REGISTER SNAPSHOTS
013260  032737  000400  002404        BIT     #DPR,@#DS1      ;TEST THAT 'DPR' = 1
013266  001411                        BEQ     9$              ;ERROR - DRIVE SHOULD BE PRESENT
013270  032737  004000  002362        BIT     #DVA,@#CS1      ;TEST THAT DEVICE IS NOW AVAILABLE
013276  001405                        BEQ     9$              ;ERROR - 'DVA' SHOULD = 1
013300  032737  000200  002404        BIT     #DRY,@#DS1      ;TEST THAT DRIVE READY IS = 1
013306  001401                        BEQ     9$              ;ERROR - 'DRY' SHOULD = 1
013310  000403                        BR      8$              ;A-OK: 'DPR' = 1, 'DVA' = 1, & 'DRY' =1
013312  104102                9$:     ERROR   102             ;UNIT NOT AVAILABLE AFTER
                                                              ;ADDRESS PLUG WAS REPLACED
013314  000137  013656                JMP     TST15   ;       ;JUMP TO NEXT TEST ----------)

013320                        8$:     ;CHANGE THE INITIAL REGISTER SNAPSHOT TO EXPECTED VALUES
                                      ;*AFTER THE PLUG CHANGE

013320  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013324  002300                        RHCS1                   ;CHANGE RHCS1 REGISTER

013326  000004                        4                       ;4 BIT/BITS TO BE CHANGED
013330  000001                        1                       ;NEW VALUE OF RDY IS 1
013332  000200                        RDY                     ;CHANGE RDY BIT
013334  000001                        1                       ;NEW VALUE OF SC IS 1
013336  100000                        SC                      ;CHANGE SC BIT
013340  000001                        1                       ;NEW VALUE OF TRE IS 1
013342  040000                        TRE                     ;CHANGE TRE BIT
013344  000001                        1                       ;NEW VALUE OF MCPE IS 1
013346  020000                        MCPE                    ;CHANGE MCPE BIT

013350  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013354  002322                        RHDS1                   ;CHANGE RHDS1 REGISTER

013356  000003                        3                       ;3 BIT/BITS TO BE CHANGED
013360  000001                        1                       ;NEW VALUE OF ATA IS 1
013362  100000                        ATA                     ;CHANGE ATA BIT
013364  000001                        1                       ;NEW VALUE OF ERR IS 1
```

```
013366  340000                          ERR                     ;CHANGE ERR BIT
013370  000000                          0                       ;NEW VALUE OF VV IS 0
013372  000100                          VV                      ;CHANGE VV BIT

013374  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013400  002276                          RHCS2                   ;CHANGE RHCS2 REGISTER

013402  000001                          1                       ;1 BIT/BITS TO BE CHANGED
013404  000001                          1                       ;NEW VALUE OF NED IS 1
013406  010000                          NED                     ;CHANGE NED BIT

013410  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013414  002314                          RHER3                   ;CHANGE RHER3 REGISTER

013416  000001                          1                       ;1 BIT/BITS TO BE CHANGED
013420  000001                          1                       ;NEW VALUE OF ACE IS 1
013422  020000                          ACE                     ;CHANGE ACE BIT

013424  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24 ;SET UNIT UNDER TEST ATTENTION
013432  005037  004654                  CLR     @#SAVERE+42     ;SET RHCC REGISTER IMAGE TO ALL 0'S

                                        ;*TAKE A NEW REGISTER SNAPSHOT AND
                                        ;*COMPARE THE REGISTER CONTENTS WITH EXPECTED VALUES


013436  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
013442  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
013444  002354                          W0                      ;TEST DATA STARTING FROM 'RHW0'
013446  000022                          18.                     ;18. REGISTERS TO BE COMPARED
013450  013454                          5$                      ;RETURN TO 5$ ON ERROR
013452  013460                          6$                      ;RETURN TO 6$ ON NO ERROR

013454  104100                  5$:     ERROR   ICJ             ;ADDRESS PLUG CHANGE CAUSED SOME
                                                                ;REGISTER ERROR
013456  000207                          RTS     PC              ;GO BACK AND CHECK THE NEXT REGISTER
                                                                ;UNTIL ALL 18. HAVE BEEN DONE
```

```
                                        ;*NOW CLEAR OUT THE CONTROLLER AND DRIVE
013460                          5$:
013460  004737  042366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER
                                        ;*CHANGE THE REGISTER SNAPSHOT TO EXPECTED VALUES AFTER THE CLEAR
013464  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013470  002300                        RHCS1                   ;CHANGE RHCS1 REGISTER
013472  000003                        3                       ;3 BIT/BITS TO BE CHANGED
013474  000000                        0                       ;NEW VALUE OF SC IS 0
013476  100000                        SC                      ;CHANGE SC BIT
013500  000000                        0                       ;NEW VALUE OF TRE IS 0
013502  040000                        TRE                     ;CHANGE TRE BIT
013504  000000                        0                       ;NEW VALUE OF MCPE IS 0
013506  020000                        MCPE                    ;CHANGE MCPE BIT
013510  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013514  002322                        RHDS1                   ;CHANGE RHDS1 REGISTER
013516  000002                        2                       ;2 BIT/BITS TO BE CHANGED
013520  000000                        0                       ;NEW VALUE OF ATA IS 0
013522  100000                        ATA                     ;CHANGE ATA BIT
013524  000000                        0                       ;NEW VALUE OF ERR IS 0
013526  040000                        ERR                     ;CHANGE ERR BIT
013530  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013534  002276                        RHCS2                   ;CHANGE RHCS2 REGISTER
013536  000001                        1                       ;1 BIT/BITS TO BE CHANGED
013540  000000                        0                       ;NEW VALUE OF NED IS 0
013542  010000                        NED                     ;CHANGE NED BIT
013544  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013550  002314                        RHER3                   ;CHANGE RHER3 REGISTER
013552  000001                        1                       ;1 BIT/BITS TO BE CHANGED
013554  000000                        0                       ;NEW VALUE OF ACE IS 0
013556  020000                        ACE                     ;CHANGE ACE BIT
013560  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
013564  002320                        RHMR                    ;CHANGE RHMR REGISTER
013566  000001                        1                       ;1 BIT/BITS TO BE CHANGED
013570  000000                        0                       ;NEW VALUE OF DMD IS 0
013572  000001                        DMD                     ;CHANGE DMD BIT
```

```
013574  043737  004740  004636          BIC     @#ATTENT,@#SAVERE+24 ;UNIT UNDER TEST ATTENTION BIT


                                        ;*TAKE ANOTHER REGISTER SNAPSHOT AND COMPARE RESULTS
                                        ;*WITH THE EXPECTED VALUES

013602  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
013606  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
013610  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
013612  000022                          18.                     ;18. REGISTERS TO BE COMPARED
013614  013620                          10$                     ;RETURN TO 10$ ON ERROR
013616  013624                          11$                     ;RETURN TO 11$ ON NO ERROR

013620  104100                  10$:    ERROR   100             ;ADDRESS PLUG CHANGE CAUSED SOME
013622  000207                          RTS     PC              ;INCORRECT REGISTER RESULT

013624                          11$:

                                        ;*(USE NED METHOD TO VERIFY
                                        ;*THAT ATTENTION BIT COMES UP IN THE PROPER LOCATION ??)



                                        ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
                                        ;*CAUSED BY PULLING THE PLUG


013624  013746  002460                  MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
013630  052716  000001                  BIS     #GO,(SP)        ;GET READY TO SET GO
                                                                ;WITHOUT INTERRUPT ENABLE
013634  012677  166440                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
                                                                ;WITH INTERRUPT DISABLED

013640  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
013642  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

013644  104412                          WAT                     ;WAIT FOR VV BIT TO SET
013646  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
013650  000100                          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
013652  000001                          1.                      ;ALLOW 10 MICRO SECONDS
013654  000001                          1.                      ;VV MUST SET BETWEEN
                                                                ;00 AND 20 MICRO SECONDS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 15        CHECK ALL ADDRESS PLUG ADDRESSES

                                        ;*      CHECK THAT ALL ADDRESS PLUG NUMBERS FUNCTION PROPERLY

                                        ;*      THIS TEST IS DONE FOR MEMOREX RP04'S AND RP06'S ONLY
                                        ;*      THIS TEST IS SELECTED BY SW#5 AND THE DEFAULT IS NOT TO DO IT

                                        ;;*****************************************************************
013656  000004                    TST15:  SCOPE
013660  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION

                                        ;;*****************************************************************
013666  005737  004744                    TST     @#RP06          ;TEST FOR RP06 DRIVE
013672  001005                            BNE     4$              ;IF = 1, OK TO DO THIS TEST
013674  005737  004746                    TST     @#RP05          ;TEST FOR MEMOREX RP04
013700  001002                            BNE     4$              ;IF = 1, OK TO DO THIS TEST
                                                                ;NOT AN RP06 OR MEMOREX RP04 SO
                                                                ;ASSUME AN ISS RP04 AND SKIP TEST
013702  000137  014664                    JMP     TST16   ;       JUMP TO NEXT TEST ------------------)
013706                              4$:
                                        ;;*****************************************************************

                                        ;*CHECK TO SEE IF THIS TEST HAS BEEN SELECTED WITH SW5

013706  032777  000040  165224            BIT     #SW5,@SWR       ;TEST THE SWITCH
013714  001002                            BNE     5$              ;IF 0, TEST HAS NOT BEEN SELECTED
013716  000137  014664                    JMP     TST16   ;       JUMP TO NEXT TEST ------------------)
013722                              5$:                             ;TEST SELECTED, CONTINUE IT

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

013722  005737  004724                    TST     @#NOPUSH        ;IS THIS A 220 START ?
013726  001007                            BNE     1$              ;SKIP THIS TEST IF SO
013730  005737  000042                    TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
013734  001004                            BNE     1$              ;SKIP THIS TEST
013736  005737  001100                    TST     @#$PASS         ;FIRST PASS ?
013742  001001                            BNE     1$              ;SKIP THIS TEST IF NOT
013744  000402                            BR      2$              ;CONTINUE WITH THIS TEST

013746                              1$:
013746  000137  014664                    JMP     TST16   ;       JUMP TO NEXT TEST ----------------)
013752                              2$:
013752  012706  001000                    MOV     #STACK,SP       ;RESET STACK
013756  012737  000015  004604            MOV     #15,@#TSTNM     ;SAVE TEST NUMBER

013764  004737  041366                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
```

```
2905  013770  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
2906                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
2907  013774  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
2908                                                                  ;THE FIRST SET OF BITS DON'T = 1
2909  014000  000000                          HALT                    ;STOP
2910
2911
2912  014002  104400  014010                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2913  014006  000426                          BR      64$             ;;GET OVER THE ASCIZ
2914                                  ;;65$:   .ASCIZ  <15><12>/ALL DISK DRIVES EXCEPTING THE UNIT UNDER/
2915  014064                          64$:
2916  014064  104400  014072                  TYPE    ,67$            ;;TYPE ASCIZ STRING
2917  014070  000417                          BR      66$             ;;GET OVER THE ASCIZ
2918                                  ;;67$:   .ASCIZ  <15><12>/TEST MUST BE POWERED DOWN/<15><12>
2919  014130                          66$:
2920
2921
2922
2923  014130  013737  004716  004752          MOV     @#UNIT,@#INUNIT ;MAKE THE INITIAL UNIT NO. = "UNIT"
2924
```

```
                                                ;*CHANGE ADDRESS PLUG ON THE UNIT UNDER TEST
014136                                  6$:
014136  104400  014144                        TYPE    ,69$            ;;TYPE ASCIZ STRING
014142  000420                              BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12>/REMOVE ADDRESS PLUG ON DRIVE/
014204                                  68$:
014204  013746  004716                        MOV     @#UNIT,-(SP)    ;GET THE UNIT UNDER TEST
014210  104404                              TYPOS                   ;TYPE IT OUT
014212  104400  014220                        TYPE    ,71$            ;;TYPE ASCIZ STRING
014216  000421                              BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  /       REPLACE WITH NEXT HIGHER ADDRESS/
014262                                  70$:
014262  104400  014270                        TYPE    ,73$            ;;TYPE ASCIZ STRING
014266  000417                              BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  <15><12>/PLUG - THEN PRESS CONTINUE/
014326                                  72$:
014326  000000                              HALT

                                                ;*HOUSEKEEPING

014330  005037  046760                        CLR     @#PRITEM        ;CLEAR THE PREVIOUS ERROR NUMBER
014334  005237  004716                        INC     @#UNIT          ;ADD ONE TO THE UNIT NO.
014340  042737  177770  004716                BIC     #^C7,@#UNIT     ;TRUNCATE TO LOW ORDER 3 BITS

014346  004737  041366                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER


                                                ;*ATTEMPT TO ADDRESS THE NEW UNIT NUMBER

014352  017700  165726                        MOV     @RHDST,R0       ;ATTEMPT TO ADDRESS THE NEW DRIVE NO.
014356  004737  043300                        JSR     PC,@#PUTREG     ;TAKE REG. SNAPSHOT IN CASE OF ERROR
014362  032737  010000  002360                BIT     #NED,@#CS2      ;TEST FOR NON EXISTENT DRIVE
014370  001423                              BEQ     7$              ;CONTINUE IF 'NED' IS NOT SET - DRIVE
                                                                        ;SHOULD BE EXISTENT ON THE BUSS
014372  104102                              ERROR   102             ;UNIT NOT AVAILABLE AFTER ADDRESS
                                                                        ;PLUG REPLACED
014374  104400  014402                        TYPE    ,75$            ;;TYPE ASCIZ STRING
014400  000407                              BR      74$             ;;GET OVER THE ASCIZ
                                        ;;75$:  .ASCIZ  <15><12>/UNIT NUMBER/
014420                                  74$:
014420  013746  004716                        MOV     @#UNIT,-(SP)    ;GET THE BAD UNIT NUMBER
014424  104404                              TYPOS                   ;TYPE IT OUT
014426  104400  014434                        TYPE    ,77$            ;;TYPE ASCIZ STRING
014432  000402                              BR      76$             ;;GET OVER THE ASCIZ
                                        ;;77$:  .ASCIZ  <15><12>/ /
014440                                  76$:
```

```
2978
2979                                        ;*CHECK IF ALL UNIT NUMBERS HAVE BEEN TRIED
2980
2981  014440  023737  004716  004752  7$:   CMP     @#UNIT,@#INUNIT ;HAVE WE INCREMENTED BACK TO THE
2982                                        ;ORIGINAL UNIT NO. YET ?
2983  014446  001233                        BNE     6$              ;NO..DO NEXT ADDRESS PLUG
2984                                        ;YES..CONTINUE WITH TESTS
2985
2986
2987
2988                                        ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
2989                                        ;*CAUSED BY PULLING THE ADDRESS PLUGS OUT
2990
2991
2992  014450  013746  002460                MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
2993  014454  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
2994                                        ;WITHOUT INTERRUPT ENABLE
2995  014460  012677  165614                MOV     (SP)+,@RHCS1    ;GO WITH
2996                                         ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
2997                                        ;WITH INTERRUPT DISABLED
2998
2999  014464  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3000  014466  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3001
3002  014470  104412                        WAT                     ;WAIT FOR VV BIT TO SET
3003  014472  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
3004  014474  000100                        VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
3005  014476  000001                        1.                      ;ALLOW 10 MICRO SECONDS
3006  014500  000001                        1.                      ;VV MUST SET BETWEEN
3007                                        ;00 AND 20 MICRO SECONDS
3008
3009  014502  104400  014510                TYPE    ,82$            ;;TYPE ASCIZ STRING
3010  014506  000436                        BR      81$             ;;GET OVER THE ASCIZ
3011                              ;;82$:    .ASCIZ  <15><12><15><12>/ALL DISK DRIVES WHICH WERE POWERED UP WHEN THE PROGRAM/
3012  014604                    81$:
3013  014604  104400  014612                TYPE    ,84$            ;;TYPE ASCIZ STRING
3014  014610  000425                        BR      83$             ;;GET OVER THE ASCIZ
3015                              ;;84$:    .ASCIZ  <15><12>/WAS STARTED MUST BE POWERED UP AGAIN/<15><12>
3016  014664                    83$:
```

```
3017
3018
3019                                    .SBTTL
3020                                    .SBTTL  **DRIVE COMMAND TESTS**
3021                                    .SBTTL
3022
3023
3024
3025
3026
3027
3028                                    ;;*****************************************************************
3029                                    ;*TEST 16       NO OPERATION FUNCTION TEST
3030                                    ;*      ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NOP"=0
3031                                    ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
3032                                    ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
3033                                    ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
3034
3035                                    ;;*****************************************************************
3036  014664  000004                    TST16:  SCOPE
3037  014666  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3038  014672  012737  000016  004604            MOV     #16,@#TSTNM     ;SAVE TEST NUMBER
3039
3040  014700  004737  041366                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3041                                                                    ;R3-RHDS1, R4-RHER1
3042                                                                    ;GIVE RH-11 INITIALIZE
3043                                                                    ;SETUP UNIT NUMBER
3044  014704  004737  041446                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3045                                                                    ;AND THAT NO STATUS BITS ARE STUCK = 1
3046  014710  104400  066402                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
3047                                                                    ;THE FIRST SET OF BITS DON'T = 1
3048  014714  000000                    HALT                            ;STOP
3049
3050  014716  013777  002422  165354    MOV     @#NOPERA,@RHCS1 ;GET READY FOR NOPERA
3051                                                            ;NO OPERATION WITH 0 IN RHCS1
3052
3053
3054                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION
3055
3056  014724  004037  041534            JSR     R0,@#SAVER      ;SAVE REGISTERS
3057  014730  002272                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3058  014732  004612                    SAVERE                  ;STARTING ADDRESS OF WHERE
3059                                                            ;THE REGISTERS ARE SAVED
3060  014734  000022                    18.                     ;NUMBER OF REGISTERS
3061                                                            ;SAVED = 18.
3062
3063  014736  013777  004606  165322    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3064                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
3065                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3066                                                            ;'TIME' WILL ONLY SAVE
3067                                                            ;CURRENT CYLINDER ADDRESS
3068                                                            ;AND LOOK AHEAD REGISTERS
3069
3070
```

```
                                                ;GET READY TO MOVE COMMAND
014744  013746  002422          MOV     @#NOFERA,-(SP)  ;GET READY TO SET GO
014750  052716  000001          BIS     #GO,(SP)        ;WITHOUT INTERRUPT ENABLE
014754  012677  165320          MOV     (SP)+,@RHCS1    ;GO WITH
                                                ;0 IN RHCS1 FOR NO-OPERATION
                                                ;WITH INTERRUPT DISABLED


014760  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
014762  003300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
014764  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
014766  000001                  1.                      ;ALLOW 10 MICRO SECONDS
014770  000001                  1.                      ;RDY MUST SET BETWEEN
                                                ;00 AND 20 MICRO SECONDS


                                ;*NOW COMPARE REGISTERS BEFORE NO-OP COMMAND
                                ;*WITH AFTER NO-OP COMMAND

014772  004037  042354          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                ;PRESENT VALUE
014776  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
015000  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
015002  000022                  18.                     ;18. REGISTERS TO BE COMPARED
015004  015010                  1$                      ;RETURN TO 1$ ON ERROR
015006  015014                  2$                      ;RETURN TO 2$ ON NO ERROR

015010  104016          1$:     ERROR   16              ;GIVING A NO-OP COMMAND
015012  000207                  RTS     PC              ;CAUSED AN ERROR
                                                ;NO REGISTERS SHOULD CHANGE
                                                ;GOOD DATA GIVES REGISTER
                                                ;CONTENTS BEFORE COMMAND
                                                ;RECEIVED DATA GIVES REGISTER
                                                ;CONTENTS AFTER COMMAND
```

```
                                        ;*NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES

015014                          2$:

015014  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
015020  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0
015024  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

015030  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA

015034  052730  043010                  BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
015040  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

015044  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

015050  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

015054  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

015060  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

015064  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA

015070  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

015074  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
015076  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
015100  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
                                                                ;AFTER ALL ATA BITS HAVE BEEN SET
015102  013700  004742                  MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
015106  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY
015110  012705  000010                  MOV     #8.,R5          ;COUNTER
015114  006000                  87$:    ROR     R0              ;GET BIT INTO CARRY
015116  103002                          BCC     88$             ;BRANCH IF NO UNIT ON THIS BIT
015120  012714  177777                  MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
                                                                ;TO SET ATA BIT
015124  005212                  88$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
015126  005305                          DEC     R5              ;COUNT
015130  001371                          BNE     87$             ;BRANCH IF 8 NOT DONE
015132  012612                          MOV     (SP)+,@R2       ;REINSTATE RHCS2
015134  012614                          MOV     (SP)+,@R4       ;REINSTATE RHER1
015136  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
015140  005720                          TST     (R0)+           ;GET OVER PHAS IN R0

015142  012730  177776                  MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
```

```
3160  015146  004737  041446                  JSR     PC,@#CHECK*     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3161                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
3162  015152  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
3163                                                                  ;THE FIRST SET OF BITS DON'T = 1
3164  015156  000000                          HALT                    ;STOP
3165  015160  013777  002422  165112          MOV     @#NOPERA,@PHCS1 ;GET READY FOR NOPERA
3166                                                                  ;NO OPERATION WITH 0 IN RHCS1
3167
3168
3169
3170                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER A NO-OP
3171  015166  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3172  015172  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3173  015174  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
3174                                                                  ;THE REGISTERS ARE SAVED
3175  015176  000022                          18.                     ;NUMBER OF REGISTERS
3176                                                                  ;SAVED = 18.
3177
3178  015200  013777  004606  165060          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3179                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3180                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3181                                                                  ;'TIME' WILL ONLY SAVE
3182                                                                  ;CURRENT CYLINDER ADDRESS
3183                                                                  ;AND LOOK AHEAD REGISTERS
3184
3185
3186  015206  013746  002422                  MOV     @#NOPERA,-(SP)  ;GET READY TO MOVE COMMAND
3187  015212  052716  000001                  BIS     #GO,(SP)        ;GET READY TO SET GO
3188                                                                  ;WITHOUT INTERRUPT ENABLE
3189  015216  012677  165056                  MOV     (SP)+,@RHCS1    ;GO WITH
3190                                                                  ;0 IN RHCS1 FOR NO-OPERATION
3191                                                                  ;WITH INTERRUPT DISABLED
3192
3193
3194
3195  015222  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
3196  015224  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
3197  015226  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3198  015230  000001                          1.                      ;ALLOW 10 MICRO SECONDS
3199  015232  000001                          1.                      ;RDY MUST SET BETWEEN
3200                                                                  ;00 AND 20 MICRO SECONDS
3201
3202                                          ;*CHANGE REGISTERS TO EXPECTED VALUES
3203
3204  015234  005737  004750                  TST     @#RH70          ;RUNNING ON AN RH70 ?
3205  015240  001406                          BEQ     5$              ;IF NOT, SKIP NEXT
3206  015242  005737  004722                  TST     @#NUNIT         ;TESTING MORE THAN ONE DRIVE ?
3207  015246  001003                          BNE     5$              ;SKIP NEXT IF SO
3208  015250  042737  100000  004620          BIC     #SC,@#SAVERE+6  ;CLEAR 'SC' IN RHCS1
3209
3210  015256                          5$:
3211
3212  015256  043737  004740  004636          BIC     @#ATTENT,@#SAVERE+24    ;CLEAR APPROPIATE ATA BITS
3213                                                                  ;FOR WORKING DRIVE IN SAVED RHAS
```

```
                                ;CHANGE BITS IN SAVED REGISTER
015264  004037  042246          JSR     R0,3$CHREG
015270  002322                  RHDS1                   ;CHANGE RHDS1 REGISTER
                                
015272  000001                  1                       ;1 BIT/BITS TO BE CHANGED
015274  000000                  0                       ;NEW VALUE OF ATA IS 0
015276  100000                  ATA                     ;CHANGE ATA BIT


                                ;*NOW COMPARE REGISTERS BEFORE NO-OP WITH
                                ;*AFTER NO-OP COMMAND

015300  004037  042354          JSR     R0,3$COMREG     ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
015304  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
015306  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
015310  000022                  18.                     ;18. REGISTERS TO BE COMPARED
015312  015316                  3$                      ;RETURN TO 3$ ON ERROR
015314  015322                  4$                      ;RETURN TO 4$ ON NO ERROR

015316  104016          3$:     ERROR   16              ;GIVING A NO-OP COMMAND
015320  000207                  RTS     PC              ;CAUSED AN ERROR
                                                        ;NO REGISTERS SHOULD CHANGE
                                                        ;GOOD DATA GIVES REGISTER
                                                        ;CONTENTS BEFORE COMMAND
                                                        ;RECEIVED DATA GIVES REGISTER
                                                        ;CONTENTS AFTER COMMAND
015322                  4$:
```

```
                                ;;*****************************************************************
                                ;*TEST 17       DRIVE CLEAR
                                ;*      ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                ;*      THEN A DRIVE CLEAR IS GIVEN
                                ;*      THEN ALL REGISTERS ARE CHECKED TO HAVE APPROPIAVE VALUE
                                ;;*****************************************************************
015322  000004                  TST17:  SCOPE
015324  012706  001000                  MOV     #STACK,SP       ;RESET STACK
015330  012737  000017  004604          MOV     #17,@#TSTNM     ;SAVE TEST NUMBER

015336  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
015342  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
015346  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
015352  000000                          HALT                    ;STOP

                                        ;*WRITE ALL WRITABLE REGISTER BITS

015354  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
015360  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

015364  012730  177776                  MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHBA

015370  052730  043010                  BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
015374  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

015400  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

015404  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

015410  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

015414  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

015420  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA

015424  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

015430  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
015432  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
015434  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
                                                                ;AFTER ALL ATA BITS HAVE BEEN SET
015436  013700  004742                  MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
015442  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY
015444  012705  000010                  MOV     #8.,R5          ;COUNTER
015450  006000              94$:        ROR     R0              ;GET BIT INTO CARRY
015452  103002                          BCC     85$             ;BRANCH IF NO UNIT ON THIS BIT
```

```
3299  015454  012714  177777                MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
3300                                                                  ;TO SET AT9 BIT
3301  015460  005212                  85$:   INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
3302  015462  005305                        DEC     R5              ;COUNT
3303  015464  001371                        BNE     94$             ;BRANCH IF 8 NOT DONE
3304  015466  012612                        MOV     (SP)+,@R2       ;REINSTATE RHCS2
3305  015470  012614                        MOV     (SP)+,@R4       ;REINSTATE RHER1
3306  015472  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
3307  015474  005720                        TST     (R0)+           ;GET OVER PHAS IN R0
3308
3309
3310  015476  012730  177776                MOV     #177776,@ R0)+  ;LOAD 177776 INTO RHMR
3311
3312
3313  015502  017737  164626  004654        MOV     @RHCC,@#SAVERE+42 ;SAVE RHCC IN SAVERE TABLE
3314  015510  013777  004606  164550        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3315                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3316                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3317                                                                  ;'TIME' WILL ONLY SAVE
3318                                                                  ;CURRENT CYLINDER ADDRESS
3319                                                                  ;AND LOOK AHEAD REGISTERS
3320
3321
3322  015516  013746  002430                MOV     @#DCLEAR,-(SP)  ;GET READY TO MOVE COMMAND
3323  015522  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
3324                                                                  ;WITHOUT INTERRUPT ENABLE
3325  015526  012677  164546                MOV     (SP)+,@RHCS1    ;GO WITH
3326                                                                  ;IO IN RHCS1 FOR DRIVE CLEAR
3327                                                                  ;WITH INTERRUPT DISABLED
3328
3329
3330
3331  015532  104412                        WAT                     ;WAIT FOR RDY BIT TO SET
3332  015534  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
3333  015536  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3334  015540  000001                        1.                      ;ALLOW 10 MICRO SECONDS
3335  015542  000001                        1.                      ;RDY MUST SET BETWEEN
3336                                                                  ;00 AND 20 MICRO SECONDS
3337
3338                                        ;*NOW LOAD 'SAVERE' REGISTER SNAPSHOT WITH EXPECTED VALUES
3339
3340  015544  004037  041270                JSR     R0,@#FILLRE     ;MOV 177777 INTO SAVED RHWC
3341  015550  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
3342  015552  177777                        177777                  ;DATA
3343  015554  004037  041270                JSR     R0,@#FILLRE     ;MOV 177776 INTO SAVED RHBA
3344  015560  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
3345  015562  177776                        177776                  ;DATA
3346  015564  005037  004616                CLR     @#SAVERE+4      ;CLEAR LOCATION FOR RHCS2
3347  015570  053737  004716  004616        BIS     @#UNIT,@#SAVERE+4;PUT UNIT # BACK IN THE SAVED RHCS2
3348
3349  015576  005737  004750                TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
3350  015602  001021                        BNE     9$              ;IF SO SKIP NEXT RH11 CODE
3351
3352
```

```
3353  015604  004037  042246          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3354  015610  002276                  RHCS2                   ;CHANGE RHCS2 REGISTER
3355
3356  015612  000003                  3                       ;3 BIT/BITS TO BE CHANGED
3357  015614  000001                  1                       ;NEW VALUE OF IR IS 1
3358  015616  000100                  IR                      ;CHANGE IR BIT
3359  015620  000001                  1                       ;NEW VALUE OF BAI IS 1
3360  015622  000010                  BAI                     ;CHANGE BAI BIT
3361  015624  000001                  1                       ;NEW VALUE OF MXF IS 1
3362  015626  001000                  MXF                     ;CHANGE MXF BIT
3363
3364  015630  004037  042246          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3365  015634  002300                  RHCS1                   ;CHANGE RHCS1 REGISTER
3366
3367  015636  000001                  1                       ;1 BIT/BITS TO BE CHANGED
3368  015640  000001                  1                       ;NEW VALUE OF SC IS 1
3369  015642  100000                  SC                      ;CHANGE SC BIT
3370  015644  000416                  BR      9$              ;SKIP NEXT RH70 CODE
3371
3372  015646                  8$:
3373
3374  015646  004037  042246          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3375  015652  002276                  RHCS2                   ;CHANGE RHCS2 REGISTER
3376
3377  015654  000002                  2                       ;2 BIT/BITS TO BE CHANGED
3378  015656  000001                  1                       ;NEW VALUE OF IR IS 1
3379  015660  000100                  IR                      ;CHANGE IR BIT
3380  015662  000001                  1                       ;NEW VALUE OF BAI IS 1
3381  015664  000010                  BAI                     ;CHANGE BAI BIT
3382  015666  005737  004722          TST     @#NUNIT         ;TESTING MORE THAN ONE DRIVE ?
3383  015672  001003                  BNE     9$              ;SKIP NEXT IF SO
3384  015674  042737  100000  004620  BIC     #SC,@#SAVERE+6  ;CLEAR 'SC' IF NOT
3385  015702                  9$:
3386  015702  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER1
3387  015706  002302                  RHER1                   ;SAVED REGISTER TO CHANGE
3388  015710  000000                  0                       ;DATA
3389  015712  004037  041270          JSR     R0,@#FILLRE     ;MOV 17437 INTO SAVED RHDST
3390  015716  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
3391  015720  017437                  17437                   ;DATA
3392  015722  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER2
3393  015726  002306                  RHER2                   ;SAVED REGISTER TO CHANGE
3394  015730  000000                  0                       ;DATA
3395  015732  004037  041270          JSR     R0,@#FILLRE     ;MOV 116000 INTO SAVED RHOF
3396  015736  002310                  RHOF                    ;SAVED REGISTER TO CHANGE
3397  015740  116000                  116000                  ;DATA
3398  015742  004037  041270          JSR     R0,@#FILLRE     ;MOV 1777 INTO SAVED RHCA
3399  015746  002312                  RHCA                    ;SAVED REGISTER TO CHANGE
3400  015750  001777                  1777                    ;DATA
3401  015752  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER3
3402  015756  002314                  RHER3                   ;SAVED REGISTER TO CHANGE
3403  015760  000000                  0                       ;DATA
3404  015762  013746  004742          MOV     @#TOTALAT,-(SP) ;GET ALL BITS OF DRIVE & PRESENT
3405                                                          ;IN RHAS
3406  015766  043716  004740          BIC     @#ATTENT,(SP)   ;CLEAR WORKING DRIVE BIT
```

```
3407  015772  012637  004636                    MOV     (SP)+,@#SAVERE+24 ;MOVE THIS INTO RHAS POSITION
3408  015776  004037  041270                    JSR     R0,@#FILLRE       ;MOV 400 INTO SAVED RHMR
3409  016002  002320                            RHMR                      ;SAVED REGISTER TO CHANGE
3410  016004  000400                            400                       ;DATA
3411
3412  016006                            3$:
3413  016006  004037  041270                    JSR     R0,@#FILLRE       ;MOV 10700 INTO SAVED RHDS1
3414  016012  002322                            RHDS1                     ;SAVED REGISTER TO CHANGE
3415  016014  010700                            10700                     ;DATA
3416
3417  016016  013737  002406  004644    4$:     MOV     @#DT,@#SAVERE+32  ;MOVE DRIVE TYPE INTO RHDT
3418                                                                      ;POSITION OF SAVRE TABLE
3419  016024  013737  002410  004646            MOV     @#SN,@#SAVERE+34  ;MOVE SERIAL NUMBER INTO RHSN
3420                                                                      ;POSITION OF SAVERE TABLE
3421
3422  016032  004037  041270                    JSR     R0,@#FILLRE       ;MOV 0 INTO SAVED RHEC1
3423  016036  002330                            RHEC1                     ;SAVED REGISTER TO CHANGE
3424  016040  000000                            0                         ;DATA
3425  016042  004037  041270                    JSR     R0,@#FILLRE       ;MOV 0 INTO SAVED RHEC2
3426  016046  002332                            RHEC2                     ;SAVED REGISTER TO CHANGE
3427  016050  000000                            0                         ;DATA
3428
3429  016052  004037  042246                    JSR     R0,@#CHREG        ;CHANGE BITS IN SAVED REGISTER
3430  016056  002300                            RHCS1                     ;CHANGE RHCS1 REGISTER
3431
3432  016060  000001                            1                         ;1 BIT/BITS TO BE CHANGED
3433  016062  000001                            1                         ;NEW VALUE OF PAR IS 1
3434  016064  000010                            PAR                       ;CHANGE PAR BIT
3435
3436                                            ;*NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
3437                                            ;*EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
3438                                            ;*WITH SAVERE TABLE
3439
3440
3441  016066  004037  042354                    JSR     R0,@#COMREG       ;COMPARE SAVED REGISTERS WITH
3442                                                                      ;PRESENT VALUE
3443  016072  004612                            SAVERE                    ;GOOD DATA SAVED IN 'SAVERE'
3444  016074  002354                            WC                        ;TEST DATA STARTING FROM 'RHWC'
3445  016076  000022                            18.                       ;18. REGISTERS TO BE COMPARED
3446  016100  016104                            5$                        ;RETURN TO 5$ ON ERROR
3447  016102  016110                            6$                        ;RETURN TO 6$ ON NO ERROR
3448
3449  016104  104017                    5$:     ERROR   17                ;DRIVE CLEAR COMMAND
3450  016106  000207                            RTS     PC                ;GAVE AN ERROR
3451                                                                      ;GOOD DATA HAS WHAT SHOULD
3452                                                                      ;BE IN REGISTER AFTER A
3453                                                                      ;DRIVE CLEAR
3454                                                                      ;RECEIVED DATA HAS WHAT
3455                                                                      ;THE REGISTER ACTUALLY
3456                                                                      ;CONTAINED
3457  016110                            6$:
3458
```

```
                                ;;*****************************************************************
                                ;*TEST 20       READ-IN-PRESET
                                ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
                                ;*
                                ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                ;*      THEN THIS TEST IS NOT PERFORMED
                                ;*
                                ;*      IF NO ACT-11 MONITOR IS PRESENT THEN
                                ;*      THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
                                ;*
                                ;*
                                ;*      THIS TESTS THE READ IN-PRESET COMMAND
                                ;*      FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO
                                ;*      RESET VV-BIT #6 IN RHDS1
                                ;*      THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                ;*      ATA FOR DRIVE UNDER TEST IS MADE = 0
                                ;*
                                ;*
                                ;*      THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
                                ;*      THEN ALL REGISTERS ARE TESTED
                                ;*      THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
                                ;*      RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
                                ;*      FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
                                ;*      ALL OTHER REGISTERS SHOULD BE UNCHANGED
                                ;;*****************************************************************
016110  000004                  TST20:  SCOPE
016112  012706  001000                  MOV     #STACK,SP       ;RESET STACK
016116  012737  000020  004604          MOV     #20,@#TSTNM     ;SAVE TEST NUMBER
                                
016124  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                
                                        ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
                                
016130  005737  004724                  TST     @#NOPUSH        ;IS THIS A 220 START ?
016134  001007                          BNE     1$              ;SKIP THIS TEST IF SO
016136  005737  000042                  TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
016142  001004                          BNE     1$              ;SKIP THIS TEST
016144  005737  001100                  TST     @#$PASS         ;FIRST PASS ?
016150  001001                          BNE     1$              ;SKIP THIS TEST IF NOT
016152  000402                          BR      2$              ;CONTINUE WITH THIS TEST
                                
016154                          1$:
016154  000137  016654                  JMP     TST21   ;       JUMP TO NEXT TEST ----------------)
016160                          2$:
016160  104400  016166                  TYPE    ,65$            ;;TYPE ASCIZ STRING
016164  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/STOP DRIVE /
```

```
3513  016204                          64$:
3514
3515  016204  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
3516  016210  104404                          TYPDS                   ;TYPE IT OUT
3517  016212  104400  001223                  TYPE    ,$CRLF
3518  016216  032713  010000          3$:     BIT     #MOL,@R3        ;MOL WILL BE HIGH TILL STOP IS HIT
3519  016222  001375                          BNE     3$              ;WAIT TILL STOP IS HIT
3520
3521  016224  104400  016232                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3522  016230  000406                          BR      66$             ;;GET OVER THE ASCIZ
3523                          ;;67$:  .ASCIZ  /START DRIVE/
3524  016246                  66$:
3525
3526  016246  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
3527  016252  104404                          TYPDS                   ;TYPE IT
3528  016254  104400  001223                  TYPE    ,$CRLF
3529
3530  016260  032713  010000          4$:     BIT     #MOL,@R3        ;MOL WILL BE LOW TILL FILE READY
3531  016264  001775                          BEQ     4$              ;WAIT TILL FILE READY
3532
3533  016266  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3534                                                                  ;R3-RHDS1, R4-RHER1
3535                                                                  ;GIVE RH-11 INITIALIZE
3536                                                                  ;SETUP UNIT NUMBER
3537  016272  004737  041424                  JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
3538                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
3539  016276  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
3540  016302  000000                          HALT                    ;STOP
3541  016304  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
3542
3543
3544                                  ;*NOW INITIALIZE ALL THE REGISTERS
3545
3546  016310  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
3547
3548  016314  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
3549
3550  016320  052730  043010                  BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
3551  016324  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
3552
3553  016330  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
3554
3555  016334  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
3556
3557  016340  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
3558
3559  016344  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF
3560
3561  016350  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
3562
3563  016354  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
3564
3565                                  ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
3566
```

```
3567  016360  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3568  016362  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
3569  016364  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
3570                                                                   ;AFTER ALL ATA BITS HAVE BEEN SET
3571  016366  013700  004742                  MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
3572  016372  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY
3573  016374  012705  000010                  MOV     #8.,R5          ;COUNTER
3574  016400  006000                  88$:    ROR     R0              ;GET BIT INTO CARRY
3575  016402  103002                          BCC     89$             ;BRANCH IF NO UNIT ON THIS BIT
3576  016404  012714  177777                  MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
3577                                                                   ;TO SET ATA BIT
3578  016410  005212                  89$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
3579  016412  005305                          DEC     R5              ;COUNT
3580  016414  001371                          BNE     88$             ;BRANCH IF 8 NOT DONE
3581  016416  012612                          MOV     (SP)+,@R2       ;REINSTATE RHCS2
3582  016420  012614                          MOV     (SP)+,@R4       ;REINSTATE RHER1
3583  016422  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3584  016424  005720                          TST     (R0)+           ;GET OVER PHAS IN R0
3585
3586
3587  016426  012730  177776                  MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
3588
3589
3590  016432  013777  004740  163656          MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE 'ATA'
3591  016440  013777  002462  163632          MOV     @#READIN,@RHCS1 ;GET READY FOR READIN
3592                                                                   ;READ IN WITH 20 IN RHCS1
3593
3594
3595                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
3596
3597  016446  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3598  016452  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3599  016454  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
3600                                                                   ;THE REGISTERS ARE SAVED
3601  016456  000022                          18.                     ;NUMBER OF REGISTERS
3602                                                                   ;SAVED = 18.
3603
3604  016460  013777  004606  163600          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3605                                                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
3606                                                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3607                                                                   ;'TIME' WILL ONLY SAVE
3608                                                                   ;CURRENT CYLINDER ADDRESS
3609                                                                   ;AND LOOK AHEAD REGISTERS
3610
3611  016466  005737  004750                  TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
3612  016472  001411                          BEQ     7$              ;SKIP NEXT FOR RH70 IF NOT
3613
3614  016474  013746  002462                  MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
3615  016500  052716  000001                  BIS     #GO,(SP)        ;GET READY TO SET GO
3616                                                                   ;WITHOUT INTERRUPT ENABLE
3617  016504  012677  163570                  MOV     (SP)+,@RHCS1    ;GO WITH
3618                                                                    ;20 IN RHCS1 FOR READ IN
3619                                                                   ;WITH INTERRUPT DISABLED
3620
```

```
3621  016510  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3622  016512  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3623  016514  000406                        BR      9$              ;SKIP NEXT FOR RH11
3624
3625  016516                          7$:
3626
3627  016516  013746  002462                MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
3628  016522  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
3629                                                                 ;ENABLE INTERRUPT
3630  016526  012677  163546                MOV     (SP)+,@RHCS1    ;GO WITH
3631                                                                  ;20 IN RHCS1 FOR READ IN
3632                                                                 ;WITH INTERRUPT ENABLED
3633
3634  016532                          8$:
3635
3636  016532  104412                        WAT                     ;WAIT FOR VV BIT TO SET
3637  016534  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
3638  016536  000100                        VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
3639  016540  000001                        1.                      ;ALLOW 10 MICRO SECONDS
3640  016542  000001                        1.                      ;VV MUST SET BETWEEN
3641                                                                 ;00 AND 20 MICRO SECONDS
3642
3643                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3644
3645  016544  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCA
3646  016550  002312                        RHCA                    ;SAVED REGISTER TO CHANGE
3647  016552  000000                        0                       ;DATA
3648  016554  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
3649  016560  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
3650  016562  000000                        0                       ;DATA
3651
3652  016564  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3653  016570  002310                        RHOF                    ;CHANGE RHOF REGISTER
3654
3655  016572  000003                        3                       ;3 BIT/BITS TO BE CHANGED
3656  016574  000000                        0                       ;NEW VALUE OF FMT22 IS 0
3657  016576  010000                        FMT22                   ;CHANGE FMT22 BIT
3658  016600  000000                        0                       ;NEW VALUE OF ECI IS 0
3659  016602  004000                        ECI                     ;CHANGE ECI BIT
3660  016604  000000                        0                       ;NEW VALUE OF HCI IS 0
3661  016606  002000                        HCI                     ;CHANGE HCI BIT
3662
3663  016610  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3664  016614  002322                        RHDS1                   ;CHANGE RHDS1 REGISTER
3665
3666  016616  000001                        1                       ;1 BIT/BITS TO BE CHANGED
3667  016620  000001                        1                       ;NEW VALUE OF VV IS 1
3668  016622  000100                        VV                      ;CHANGE VV BIT
3669
3670                                        ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
3671                                        ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
3672                                        ;*COMPARISONS ARE MADE
3673
3674
```

```
3675  016624  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3676                                                                  ;PRESENT VALUE
3677  016630  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3678  016632  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3679  016634  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3680  016636  016642                          5$                      ;RETURN TO 5$ ON ERROR
3681  016640  016646                          6$                      ;RETURN TO 6$ ON NO ERROR
3682
3683  016642  104020                  5$:     ERROR   20              ;READ IN COMMAND GAVE AN
3684  016644  000207                          RTS     PC              ;ERROR
3685                                                                  ;GOOD DATA HAS WHAT SHOULD
3686                                                                  ;BE IN REGISTER AFTER A
3687                                                                  ;READ-IN COMMAND
3688                                                                  ;RECEIVED DATA HAS WHAT
3689                                                                  ;THE REGISTER ACTUALLY CONTAINED
3690                                                                  ;THE FOLLOWING SHOULD
3691                                                                  ;BE THE REGISTER CONTENTS
3692                                                                  ;RHCA=0, RHDST = 0
3693                                                                  ;RHOF SHOULD HAVE FMT22 = 0,
3694                                                                  ;HCI = 0, ECI = 0,
3695                                                                  ;RHDS1 SHOULD HAVE VV = 1
3696                                                                  ;ALL OTHER BITS SHOULD
3697                                                                  ;BE UNCHANGED
3698  016646  012737  177777  046760  6$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
```

```
                        ;;*****************************************************************
                        ;*TEST 21        READ-IN-PRESET
                        ;*
                        ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT
                        ;*      THAT IT DOES NOT TEST THE SETTING OF VV
                        ;*
                        ;*      THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT
                        ;*      THEN THE PREVIOUS TEST WILL NOT BE PERFORMED
                        ;*      THIS TESTS THE READ-IN-PRESET COMMAND
                        ;*      ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                        ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                        ;*      ATA FOR DRIVE UNDER TEST IS MADE = 0
                        ;*
                        ;*      THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
                        ;*      THEN ALL REGISTERS ARE TESTED
                        ;*      THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
                        ;*      RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
                        ;*      FMT22 = 0, ECI = 0, HCI = 0, RHDSI SHOULD HAVE VV = 1
                        ;*      ALL OTHER REGISTERS SHOULD BE UNCHANGED
                        ;;*****************************************************************
016654  000004         TST21:  SCOPE
016656  012706  001000          MOV     #STACK,SP       ;RESET STACK
016662  012737  000021  004634  MOV     #21.,@#TSTNM    ;SAVE TEST NUMBER

016670  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
016674  004737  041424          JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
016700  104400  066402          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
016704  000000                  HALT                    ;STOP
016706  012700  002272          MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0


                                ;*INITIALIZE ALL THE REGISTERS

016712  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

016716  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA

016722  052730  043010          BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
016726  012730  001400          MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

016732  012730  000000          MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

016736  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

016742  012730  000000          MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

016746  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF
```

```
                                                                                                                                
                                                                                                                                
016752  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
                                                                                                                                
016756  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
                                                                                                                                
016762  012730  177777                MOV     #-1,@(R0)+      ;CLEAR ALL BITS OF RHAS
016766  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
                                                                                                                                
                                                                                                                                
016772  013777  004740  163316        MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE 'ATA'
017000  013777  002462  163272        MOV     @#READIN,@RHCS1 ;GET READY FOR READIN
                                                              ;READ IN WITH 20 IN RHCS1
                                                                                                                                
                                                                                                                                
                                                              ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
                                                                                                                                
017006  004537  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
017012  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
017014  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                              ;THE REGISTERS ARE SAVED
017016  000022                        18.                     ;NUMBER OF REGISTERS
                                                              ;SAVED = 18.
                                                                                                                                
017020  013777  004606  163240        MOV     @#RP4VEC,@PPVEC ;SET RPC4 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
                                                                                                                                
017026  005737  004750                TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
017032  001411                        BEQ     9$              ;SKIP NEXT IF NOT
                                                                                                                                
017034  013746  002462                MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
017040  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
                                                              ;WITHOUT INTERRUPT ENABLE
017044  012677  163230                MOV     (SP)+,@RHCS1    ;GO WITH
                                                               ;20 IN RHCS1 FOR READ IN
                                                              ;WITH INTERRUPT DISABLED
                                                                                                                                
                                                                                                                                
017050  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
017052  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
017054  000406                        BR      10$             ;SKIP NEXT RH11 CODE
                                                                                                                                
017056                          9$:                                                                                           
                                                                                                                                
017056  013746  002462                MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
017062  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                              ;ENABLE INTERRUPT
017066  012677  163206                MOV     (SP)+,@RHCS1    ;GO WITH
                                                               ;20 IN RHCS1 FOR READ IN
                                                              ;WITH INTERRUPT ENABLED
                                                                                                                                
017072                          10$:
```

```
 3807
 3808  017072  104412                    WAT                     ;WAIT FOR RDY BIT TO SET
 3809  017074  002300                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
 3810  017076  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 3811  017100  000001                    1.                      ;ALLOW 10 MICRO SECONDS
 3812  017102  000001                    1.                      ;RDY MUST SET BETWEEN
 3813                                                            ;00 AND 20 MICRO SECONDS
 3814
 3815                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
 3816
 3817  017104  004037  041270            JSR     R0,3$*FILLRE    ;MOV 0 INTO SAVED RHCA
 3818  017110  002312                    RHCA                    ;SAVED REGISTER TO CHANGE
 3819  017112  000000                    0                       ;DATA
 3820  017114  004037  041270            JSR     R0,3$*FILLRE    ;MOV 0 INTO SAVED RHDST
 3821  017120  002304                    RHDST                   ;SAVED REGISTER TO CHANGE
 3822  017122  000000                    0                       ;DATA
 3823
 3824  017124  004037  042246            JSR     R0,3$*CHREG     ;CHANGE BITS IN SAVED REGISTER
 3825  017130  002310                    RHOF                    ;CHANGE RHOF REGISTER
 3826
 3827  017132  000003                    3                       ;3 BIT/BITS TO BE CHANGED
 3828  017134  000000                    0                       ;NEW VALUE OF FMT22 IS 0
 3829  017136  010000                    FMT22                   ;CHANGE FMT22 BIT
 3830  017140  000000                    0                       ;NEW VALUE OF ECI IS 0
 3831  017142  004000                    ECI                     ;CHANGE ECI BIT
 3832  017144  000000                    0                       ;NEW VALUE OF HCI IS 0
 3833  017146  002000                    HCI                     ;CHANGE HCI BIT
 3834
 3835  017150  004037  042246            JSR     R0,3$*CHREG     ;CHANGE BITS IN SAVED REGISTER
 3836  017154  002322                    RHDS1                   ;CHANGE RHDS1 REGISTER
 3837
 3838  017156  000001                    1                       ;1 BIT/BITS TO BE CHANGED
 3839  017160  000001                    1                       ;NEW VALUE OF VV IS 1
 3840  017162  000100                    VV                      ;CHANGE VV BIT
 3841
 3842                                    ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
 3843                                    ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
 3844                                    ;*COMPARISONS ARE MADE
 3845
 3846
 3847  017164  004037  042354            JSR     R0,3$*COMREG    ;COMPARE SAVED REGISTERS WITH
 3848                                                            ;PRESENT VALUE
 3849  017170  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 3850  017172  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
 3851  017174  000022                    18.                     ;18. REGISTERS TO BE COMPARED
 3852  017176  017202                    5$                      ;RETURN TO 5$ ON ERROR
 3853  017200  017206                    6$                      ;RETURN TO 6$ ON NO ERROR
 3854
 3855  017202  104020            5$:     ERROR   20              ;READ-IN COMMAND GAVE AN
 3856  017204  000207                    RTS     PC              ;ERROR
 3857                                                            ;GOOD DATA HAS WHAT SHOULD
 3858                                                            ;BE IN REGISTER AFTER A
 3859                                                            ;READ-IN COMMAND
 3860                                                            ;RECEIVED DATA HAS WHAT
```

```
                                                        ;THE REGISTER ACTUALLY CONTAINE
                                                        ;THE FOLLOWING SHOULD
                                                        ;BE THE REGISTER CONTENTS
                                                        ;RHCA = 0, RHDST = 0
                                                        ;RHOF SHOULD HAVE FMT22 = 0,
                                                        ;HCI = 0, ECI = 0,
                                                        ;RHDS1 SHOULD HAVE VV = 1
                                                        ;ALL OTHER BITS SHOULD
                                                        ;BE UNCHANGED
017236                          63:
```

```
                                ;;*****************************************************************
017206  005737  004744                TST     @#RPO6          ;TEST FOR RPO6 DRIVE
017212  001401                        BEQ     7$              ;IF = 0, TREAT DRIVE AS AN RPO4
017214  000402                        BR      8$              ;TREAT AS RPO6 - DO NEXT "MAKECL"
017216  000137  017262          7$:   JMP     @#DOG           ;DO SECOND FOLLOWING "MAKECL"
017222                          8$:
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 22       MAKE CURRENT CYLINDER = 777
                                ;;*****************************************************************
017222  000004                  TST22:  SCOPE
017224  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017230  012737  000022  004604          MOV     #23-1,@#TSTNM   ;THIS SAVES TEST NUMBER

017236  004737  041366                  JSR     PC,@#CLDISK     ;INIT DRIVE
017242  012777  000001  163050          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
017250  004037  041036                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLLOWED BY A INIT
                                                                ;THIS SHOULD CHANGE RHCC TO 777
017254  000777                          777

017256  000137  017316                  JMP     @#FISH          ;DON'T DO NEXT "MAKECL"

017262                          DOG:

                                ;;*****************************************************************
                                ;*TEST 23       MAKE CURRENT CYLINDER = 377
                                ;;*****************************************************************
017262  000004                  TST23:  SCOPE
017264  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017270  012737  000023  004604          MOV     #24-1,@#TSTNM   ;THIS SAVES TEST NUMBER

017276  004737  041366                  JSR     PC,@#CLDISK     ;INIT DRIVE
017302  012777  000001  163010          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
017310  004037  041036                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLLOWED BY A INIT
                                                                ;THIS SHOULD CHANGE RHCC TO 377
017314  000377                          377
```

```
                                FISH:
                                ;;*****************************************************************
                                ;*TEST 24        RECALIBRATE COMMAND

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

                                ;*      NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0

                                ;;*****************************************************************
017316  000004                TST24:  SCOPE
017320  012706  001000                MOV     #STACK,SP       ;RESET STACK
017324  012737  000024  004604        MOV     #24.,@#TSTNM    ;SAVE TEST NUMBER

017332  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER
017336  004737  041446                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
017342  104400  066402                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                              ;THE FIRST SET OF BITS DON'T = 1
017346  000000                        HALT                    ;STOP
017350  012700  002272                MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
017354  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

017360  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHBA

017364  052730  000010                BIS     #010,@(R0)+     ;LOAD 010 INTO RHCS2
017370  012730  001400                MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

017374  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

017400  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

017404  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

017410  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

017414  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA

017420  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

                                ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

017424  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
017426  011446                        MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
017430  011246                        MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
                                                              ;AFTER ALL ATA BITS HAVE BEEN SET
017432  013700  004742                MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
017436  005012                        CLR     @R2             ;CLEAR RHCS2 AND CARRY
```

```
3970  017440  012705  000010          MOV     #8.,R5          ;COUNTER
3971  017444  006000          84$:    ROR     R0              ;GET BIT INTO CARRY
3972  017446  103002                  BCC     95$             ;BRANCH IF NO UNIT ON THIS BIT
3973  017450  012714  177777          MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
3974                                                          ;TO SET ATA BIT
3975  017454  005212          95$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
3976  017456  005305                  DEC     R5              ;COUNT
3977  017460  001371                  BNE     84$             ;BRANCH IF 8 NOT DONE
3978  017462  012612                  MOV     (SP)+,@R2       ;REINSTATE RHCS2
3979  017464  012614                  MOV     (SP)+,@R4       ;REINSTATE RHER1
3980  017466  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
3981  017470  005720                  TST     (R0)+           ;GET OVER PHAS IN R0
3982
3983
3984  017472  012730  177776          MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
3985
3986
3987  017476  013777  002426  162574  MOV     @#RECALI,@RHCS1 ;GET READY FOR RECALI
3988                                                          ;RECALIBRATE WITH 6 IN RHCS1
3989
3990
3991                                  ;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
3992
3993  017504  004037  041534          JSR     R0,@#SAVER      ;SAVE REGISTERS
3994  017510  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3995  017512  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
3996                                                          ;THE REGISTERS ARE SAVED
3997  017514  000022                  18.                     ;NUMBER OF REGISTERS
3998                                                          ;SAVED = 18.
3999  017516  013777  004606  162542  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4000                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
4001                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4002                                                          ;'TIME' WILL ONLY SAVE
4003                                                          ;CURRENT CYLINDER ADDRESS
4004                                                          ;AND LOOK AHEAD REGISTERS
4005
4006  017524  013746  002426          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
4007  017530  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
4008                                                          ;ENABLE INTERRUPT
4009  017534  012677  162540          MOV     (SP)+,@RHCS1    ;GO WITH
4010                                                          ;6 IN RHCS1 FOR RECALIBRATE
4011                                                          ;WITH INTERRUPT ENABLED
4012  017540  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
4013  017542  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
4014
4015  017544  104412                  WAT                     ;WAIT FOR DRY BIT TO SET
4016  017546  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
4017  017550  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4018  017552  076377                  31999.                  ;ALLOW 319990 MICRO SECONDS
4019  017554  056701                  24001.                  ;DRY MUST SET BETWEEN
4020                                                          ;79980 AND 560000 MICRO SECONDS
4021
4022                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4023                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
```

```
4034
4035  017556  013746  002426                  MOV     @#RECALI,-(SP)  ;SAVE COMMAND
4036  017562  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
4037  017566  005737  004722                  TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
4038  017572  001413                          BEQ     89$             ;BRANCH IF ONLY ONE UNIT
4039  017574  010037  004760                  MOV     R0,@#TMP4       ;GET RHCS1
4040  017600  042737  177677  004760          BIC     #^CIE,@#TMP4    ;KEEP IE BIT
4041  017606  042716  000100                  BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
4042  017612  053716  004760                  BIS     @#TMP4,(SP)     ;GET IE AS IS
4043  017616  052716  100000                  BIS     #SC,(SP)        ;SET SC IN RHCS1
4044  017622                          89$:
4045  017622  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4046  017626  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
4047                                                                  ;AND COMMAND SHOULD BE SET
4048  017630  001405                          BEQ     88$             ;BRANCH IF GOOD
4049  017632  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
4050  017636  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
4051  017642  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
4052                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4053  017644  012746  030500          88$:    MOV     #MOL!DPR!VV!PIP,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
4054  017650  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4055  017654  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
4056                                                                  ;SHOULD BE SET
4057  017656  001405                          BEQ     90$             ;BRANCH IF GOOD
4058  017660  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
4059  017664  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
4060  017670  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
4061                                                                  ;MOL!DPR!VV!PIP SHOULD BE SET
4062  017672                          90$:
4063
4064                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4065
4066  017672  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
4067  017676  002334                          RHCC                    ;SAVED RFGISTER TO CHANGE
4068  017700  000000                          0                       ;DATA
4069  017702  004037  041270                  JSR     R0,@#FILLRE     ;MOV 116377 INTO SAVED RHOF
4070  017706  002310                          RHOF                    ;SAVED REGISTER TO CHANGE
4071  017710  116377                          116377                  ;DATA
4072
4073  017712  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
4074                                                                  ;FOR WORKING DRIVE IN
4075                                                                  ;SAVED RHAS LOACTION
4076  017720  004037  041270                  JSR     R0,@#FILLRE     ;MOV 104206 INTO SAVED RHCS1
4077  017724  002300                          RHCS1                   ;SAVED REGISTER TO CHANGE
4078  017726  104206                          104206                  ;DATA
4079  017730  004037  041270                  JSR     R0,@#FILLRE     ;MOV 110700 INTO SAVED RHDS1
4080  017734  002322                          RHDS1                   ;SAVED REGISTER TO CHANGE
4081  017736  110700                          110700                  ;DATA
4082
4083
4084                                          ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4085
4086
4087  017740  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
4078
4079  017744  004612                          SAVERE                  ;PRESENT VALUE
4080  017746  002354                          WC                      ;GOOD DATA SAVED IN 'SAVERE'
4081  017750  000022                          18.                     ;TEST DATA STARTING FROM 'RHWC'
4082  017752  017756                          1$                      ;18. REGISTERS TO BE COMPARED
4083  017754  017762                          2$                      ;RETURN TO 1$ ON ERROR
4084                                                                  ;RETURN TO 2$ ON NO ERROR
4085  017756  104064              1$:         ERROR   64              ;RECALIBRATE COMMAND CAUSED
4086  017760  000207                          RTS     PC              ;AN ERROR
4087                                                                  ;GOOD DATA GIVES WHAT SHOULD
4088                                                                  ;BE THERE
4089                                                                  ;RECEIVED DATA GIVES WHAT WAS
4090                                                                  ;THERE AFTER COMMAND
4091:
4092  017762              2$:
```

```
4093
4094                                    ;;*****************************************************************
4095  017762  005737  004744                      TST     @#RP06          ;TEST FOR RP06 DRIVE
4096  017766  001401                              BEQ     3$              ;IF = 0, TREAT DRIVE AS AN RP04
4097  017770  000402                              BR      4$              ;TREAT AS RP06 - DO NEXT "MAKECL"
4098  017772  000137  020036              3$:     JMP     @#CAT           ;DO SECOND FOLLOWING "MAKECL"
4099  017776                              4$:
4100                                    ;;*****************************************************************
4101
4102
4103
4104                                    ;;*****************************************************************
4105                                    ;*TEST 25       MAKE CURRENT CYLINDER = 777
4106                                    ;;*****************************************************************
4107  017776  000004                    TST25:  SCOPE
4108  020000  012706  001000                      MOV     #STACK,SP       ;RESET STACK
4109  020004  012737  000025  004604              MOV     #26-1,@#TSTNM   ;THIS SAVES TEST NUMBER
4110
4111  020012  004737  0 1366                      JSR     PC,@#CLDISK     ;INIT DRIVE
4112  020016  012777  000001  162274              MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
4113  020024  004037  041036                      JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
4114                                                                      ;COMMAND FOLLOWED BY A INIT
4115                                                                      ;THIS SHOULD CHANGE RHCC TO 777
4116  020030  000777                              777
4117
4118  020032  000137  020072                      JMP     @#BIRD          ;DON'T DO NEXT "MAKECL"
4119
4120
4121  020036                            CAT:
4122
4123                                    ;;*****************************************************************
4124                                    ;*TEST 26       MAKE CURRENT CYLINDER = 377
4125                                    ;;*****************************************************************
4126  020036  000004                    TST26:  SCOPE
4127  020040  012706  001000                      MOV     #STACK,SP       ;RESET STACK
4128  020044  012737  000026  004604              MOV     #27-1,@#TSTNM   ;THIS SAVES TEST NUMBER
4129
4130  020052  004737  041366                      JSR     PC,@#CLDISK     ;INIT DRIVE
4131  020056  012777  000001  162234              MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
4132  020064  004037  041036                      JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
4133                                                                      ;COMMAND FOLLOWED BY A INIT
4134                                                                      ;THIS SHOULD CHANGE RHCC TO 377
4135  020070  000377                              377
4136
```

```
                                BIRD:
                                ;;*****************************************************************
                                ;*TEST 27       RECALIBRATE COMMAND
                                ;
                                ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH 0
                                ;*      THEN A RECALIBRATE =6 COMMAND IS GIVEN
                                ;
                                ;*      NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0
                                ;
                                ;;*****************************************************************
020072  000004                TST27:  SCOPE
020074  012706  001000                MOV     #STACK,SP       ;RESET STACK
020100  012737  000027  004604        MOV     #27,@#TSTNM     ;SAVE TEST NUMBER
                                ;
020106  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER
020112  004737  041446                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
020116  104400  056402                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                              ;THE FIRST SET OF BITS DON'T = 1
020122  000000                        HALT                    ;STOP
020124  012700  002272                MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0
                                ;
020130  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHWC
                                ;
020134  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHBA
                                ;
020140  052730  000000                BIS     #0,@(R0)+       ;LOAD 0 INTO RHCS2
020144  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHCS1
                                ;
020150  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
                                ;
020154  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHDST
                                ;
020160  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
                                ;
020164  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHOF
                                ;
020170  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHCA
                                ;
020174  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
                                ;
020200  012730  177777                MOV     #-1,@(R0)+      ;CLEAR ALL BITS OF RHAS
020204  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHMR
                                ;
020210  013777  002426  162062        MOV     @#RECALI,@RHCS1 ;GET READY FOR RECALI
                                                              ;RECALIBRATE WITH 6 IN RHCS1
```

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

020216  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
020222  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
020224  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
020226  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.
020230  013777  004606  162030          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

020236  013746  002426                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
020242  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
020246  012677  162026                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;6 IN RHCS1 FOR RECALIBRATE
                                                                ;WITH INTERRUPT ENABLED
020252  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
020254  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

020256  104412                          WAT                     ;WAIT FOR DRY BIT TO SET
020260  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
020262  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
020264  076377                          31999.                  ;ALLOW 319990 MICRO SECONDS
020266  056701                          24001.                  ;DRY MUST SET BETWEEN
                                                                ;79980 AND 560000 MICRO SECONDS

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

020270  013746  002426                  MOV     @#RECALI,-(SP)  ;SAVE COMMAND
020274  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
020300  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
020304  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
                                                                ;AND COMMAND SHOULD BE SET
020306  001405                          BEQ     88$             ;BRANCH IF GOOD
020310  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
020314  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
020320  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
020322  012746  030500          88$:    MOV     #MOL!DPR!VV!PIP,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
020326  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
020332  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
                                                                ;SHOULD BE SET
020334  001405                          BEQ     90$             ;BRANCH IF GOOD
020336  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
020342  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
020346  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV!PIP SHOULD BE SET
020350                          90$:
```

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

020350  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
020354  002334                          RHCC                    ;SAVED REGISTER TO CHANGE
020356  000000                          0                       ;DATA

020360  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
                                                                ;FOR WORKING DRIVE IN
                                                                ;SAVED RHAS LOACTION

020366  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020372  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER

020374  000001                          1                       ;1 BIT/BITS TO BE CHANGED
020376  000001                          1                       ;NEW VALUE OF ATA IS 1
020400  100000                          ATA                     ;CHANGE ATA BIT

020402  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020406  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER

020410  000001                          1                       ;1 BIT/BITS TO BE CHANGED
020412  000001                          1                       ;NEW VALUE OF SC IS 1
020414  100000                          SC                      ;CHANGE SC BIT

                                        ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND

020416  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
020422  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
020424  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
020426  000022                          18.                     ;18. REGISTERS TO BE COMPARED
020430  020434                          1$                      ;RETURN TO 1$ ON ERROR
020432  020440                          2$                      ;RETURN TO 2$ ON NO ERROR

020434  104064          1$:             ERROR   64              ;RECALIBRATE COMMAND CAUSED
020436  000207                          RTS     PC              ;AN ERROR
                                                                ;GOOD DATA GIVES WHAT SHOULD BE
                                                                ;THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
020440                  2$:                                     ;THERE AFTER A RECALIBRATE
```

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

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

                                ;;*****************************************************************
020440  000004                  TST30:  SCOPE


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

020442  005737  004724                  TST     @#NOPUSH        ;IS THIS A 220 START ?
020446  001007                  BNE     1$              ;SKIP THIS TEST IF SO
020450  005737  000042                  TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
020454  001004                  BNE     1$              ;SKIP THIS TEST
020456  005737  001100                  TST     @#$PASS         ;FIRST PASS ?
020462  001001                  BNE     1$              ;SKIP THIS TEST IF NOT
020464  000402                  BR      2$              ;CONTINUE WITH THIS TEST

020466                  1$:
020466  000137  021754          JMP     TST31   ;       JUMP TO NEXT TEST ---------------->
020472                  2$:
020472  012706  001000          MOV     #STACK,SP       ;RESET STACK
020476  012737  000030  004634  MOV     #30,@#TSTNM     ;SAVE TEST NUMBER

020504  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
020510  004737  041446          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
020514  104400  066402          TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
020520  000000                  HALT                    ;STOP

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

020522  017746  161574          MOV     @RHDS1,-(SP)    ;PUSH RHDS1 ONTO STACK
```

```
 4341  020526  042716  167677                 BIC     #167677,(SP)    ;CLEAR EVERYTHING EXCEPT VV AND MOL
 4342  020532  022726  010100                 CMP     #VV!MOL,(SP)+   ;ARE VV AND MOL SET ?
 4343  020536  001504                         BEQ     6$              ;CONTINUE IF YES
 4344  020540  104400  020546                 TYPE    ,65$            ;;TYPE ASCIZ STRING
 4345  020544  000427                         BR      64$             ;;GET OVER THE ASCIZ
 4346                                 ;;65$:  .ASCIZ  <15><12>/GET DRIVE HEADS LOADED THEN HIT "CONTINUE"/
 4347  020624                         64$:
 4348  020624  104400  020632                 TYPE    ,67$            ;;TYPE ASCIZ STRING
 4349  020630  000424                         BR      66$             ;;GET OVER THE ASCIZ
 4350                                 ;;67$:  .ASCIZ  <15><12>/IF ALREADY LOADED THEN HIT "CONTINUE"/
 4351  020702                         66$:
 4352  020702  000000                         HALT                    ;WAIT FOR CONTINUE
 4353
 4354
 4355  020704  004737  041366                 JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 4356                                                                  ;R3-RHDS1, R4-RHER1
 4357                                                                  ;GIVE RH-11 INITIALIZE
 4358                                                                  ;SETUP UNIT NUMBER
 4359  020710  004737  041424                 JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
 4360                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
 4361  020714  104400  066402                 TYPE    .CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
 4362  020720  000000                         HALT                    ;STOP
 4363
 4364                                         ;*SET VV IN RHDS1 WITH PACK ACKNOWLEDGE
 4365
 4366  020722  013746  002460                 MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
 4368  020726  052716  000001                 BIS     #GO,(SP)        ;GET READY TO SET GO
 4369                                                                  ;WITHOUT INTERRUPT ENABLE
 4370  020732  012677  161342                 MOV     (SP)+,@RHCS1    ;GO WITH
 4371                                                                  ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
 4372                                                                  ;WITH INTERRUPT DISABLED
 4373
 4374
 4375
 4376  020736  104412                         WAT                     ;WAIT FOR VV BIT TO SET
 4377  020740  002322                         RHDS1                   ;WAIT FOR RHDS1 REGISTER
 4378  020742  000100                         VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
 4379  020744  000001                         1.                      ;ALLOW 10 MICRO SECONDS
 4380  020746  000001                         1.                      ;VV MUST SET BETWEEN
 4381                                                                  ;00 AND 20 MICRO SECONDS
 4382
 4383  020750                         65:
 4384  020750  004737  041446                 JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
 4385  020754  000240                         NOP                     ;CHECK THAT ALL OTHER BITS = 0
 4386  020756  000240                         NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
 4387  020760  000240                         NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
 4388                                                                  ;USED IN THE MIDDLE OF A SINGLE TEST
 4389
 4390  020762  012700  002272                 MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0
 4391
 4392                                         ;*LOAD ALL POSSIBLE REGISTERS WITH ONES
 4393
 4394
```

```
4395  020766  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
4396
4397
4398
4399  020772  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
4400
4401
4402
4403  020776  052730  000010          BIS     #10,@(R0)+      ;LOAD 10 INTO RHCS2
4404
4405
4406  021002  012730  001400          MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
4407
4408
4409
4410  021006  012730  000000          MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
4411
4412
4413
4414  021012  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
4415
4416
4417
4418  021016  012730  000000          MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
4419
4420
4421
4422  021022  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF
4423
4424
4425
4426  021026  012730  177777          MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
4427
4428
4429
4430  021032  012730  000000          MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
4431
4432
4433
4434                                  ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
4435
4436  021036  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4437  021040  011446                  MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
4438  021042  011245                  MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
4439                                                          ;AFTER ALL ATA BITS HAVE BEEN SET
4440  021044  013700  004742          MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
4441  021050  005012                  CLR     @R2             ;CLEAR RHCS2 AND CARRY
4442  021052  012705  000010          MOV     #8.,R5          ;COUNTER
4443  021056  006000          91$:    ROR     R0              ;GET BIT INTO CARRY
4444  021060  103002                  BCC     92$             ;BRANCH IF NO UNIT ON THIS BIT
4445  021062  012714  177777          MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
4446                                                          ;TO SET ATA BIT
4447  021066  005212          92$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
4448  021070  005305                  DEC     R5              ;COUNT
```

```
4449  021072  001371                        BNE     91$             ;BRANCH IF 8 NOT DONE
4450  021074  012612                        MOV     (SP)+,@R2       ;REINSTATE RHCS2
4451  021076  012614                        MOV     (SP)+,@R4       ;REINSTATE RHER1
4452  021100  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
4453  021102  005720                        TST     (R0)+           ;GET OVER PHAS IN R0
4454
4455
4456
4457
4458  021104  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMP
4459
4460
4461  021110  013777  002424  161162        MOV     @#UNLOAD,@RHCS1 ;GET READY FOR UNLOAD
4462                                                                ;UNLOAD WITH 2 IN RHCS1
4463
4464
4465                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER UNLOAD
4466
4467  021116  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
4468  021122  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4469  021124  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
4470                                                                ;THE REGISTERS ARE SAVED
4471  021126  000021                        17.                     ;NUMBER OF REGISTERS
4472                                                                ;SAVED = 17.
4473
4474  021130  013777  004606  161130        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4475                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
4476                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4477                                                                ;'TIME' WILL ONLY SAVE
4478                                                                ;CURRENT CYLINDER ADDRESS
4479                                                                ;AND LOOK AHEAD REGISTERS
4480
4481
4482  021136  013746  002424                MOV     @#UNLOAD,-(SP)  ;GET READY TO MOVE COMMAND
4483  021142  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
4484                                                                ;WITHOUT INTERRUPT ENABLE
4485  021146  012677  161126                MOV     (SP)+,@RHCS1    ;GO WITH
4486                                                                ;2 IN RHCS1 FOR UNLOAD
4487                                                                ;WITH INTERRUPT DISABLED
4488
4489  021152  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
4490  021154  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
4491
4492  021156  104412                        WAT                     ;WAIT FOR BIT TO SET
4493  021160  002300                        RHCS1                   ;IN RHCS1 REGISTER
4494  021162  000200                        RDY                     ;'RDY' BIT
4495  021164  000001                        1.                      ;ALLOW 10 MICRO SECONDS
4496  021166  000001                        1.                      ;ANOTHER 10 - 'RDY' MUST SET BETWEEN
4497                                                                ;00 AND 20 MICRO SECONDS
```

```
                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1, WHICH WERE SAVED
                                        ;*DURING THE UNLOAD COMMAND, WITH THE EXPECTED RESULTS

021170 013746 002424                    MOV     @#UNLOAD,-(SP)  ;PUSH COMMAND ON STACK
021174 052716 004201                    BIS     #DVA!GO!RDY,(SP) ;INCLUDE THESE BITS SET
021200 005737 004722                    TST     @#NUNIT         ;IS THERE MORE THAN ONE UNIT ?
021204 001413                           BEQ     9$              ;SKIP NEXT IF ONLY ONE UNIT
021206 010037 004760                    MOV     R0,@#TMP4       ;PUT SAVED RHCS1 INTO TMP4
021212 042737 177677 004760             BIC     #^C<IE>,@#TMP4  ;MASK ALL BUT THE 'IE' BIT IN RHCS1
021220 042716 000100                    BIC     #IE,(SP)        ;CLEAR 'IE' IN EXPECTED DATA
021224 053716 004760                    BIS     @#TMP4,(SP)     ;SET 'IE' STATE FROM ACTUAL RHCS1 DATA
021230 052716 100000                    BIS     #SC,(SP)        ;SET 'SC' IN RHCS1 SAVED DATA

021234 011637 001124            9$:     MOV     (SP),@#$GDDAT   ;SAVE EXPECTED DATA FOR PRINTOUT
021240 022600                           CMP     (SP)+,R0        ;COMPARE EXPECTED DATA WITH SAVED
                                                                ;RHCS1 DATA AND RESET THE STACK
021242 001405                           BEQ     10$             ;CHECK NEXT BITS IF THESE OK
021244 010037 001126                    MOV     R0,@#$BDDAT     ;RHCS1 IS BAD - PRINT IT OUT
021250 010137 004600                    MOV     R1,@#REGADR     ;REGISTER ADDRESS
021254 104021                           ERROR   21              ;DURING ABOVE OPERATION ONLY THE
                                                                ;'DVA', 'GO', 'RDY' AND COMMAND BITS
                                                                ;SHOULD BE SET

021256 012746 020400            10$:    MOV     #PIP!DPR,-(SP)  ;PUT SOME EXPECTED RHDS1 BITS ON STACK
021262 010537 004760                    MOV     R5,@#TMP4       ;PUT SAVED RHDS1 INTO TMP4
021266 042737 167677 004760             BIC     #^C<MOL!VV>,@#TMP4 ;MASK ALL BUT 'MOL' & 'VV' IN RHDS1
021274 042716 010100                    BIC     #MOL!VV,(SP)    ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
021300 053716 004760                    BIS     @#TMP4,(SP)     ;SET EXPECTED 'MOL' & 'VV' BIT STATES
                                                                ;FROM THE ACTUAL DATA (DON'T CARE)
021304 011637 001124                    MOV     (SP),@#$GDDAT   ;SAVE EXPECTED DATA FOR PRINTOUT
021310 022605                           CMP     (SP)+,R5        ;COMPARE EXPECTED DATA WITH SAVED
                                                                ;RHDS1 DATA AND RESET THE STACK
021312 001405                           BEQ     11$             ;CONTINUE IF EXPECTED=SAVED
021314 010537 001126                    MOV     R5,@#$BDDAT     ;RHDS1 IS BAD - PRINT IT OUT
021320 010337 004600                    MOV     R3,@#REGADR     ;REGISTER ADDRESS
021324 104063                           ERROR   63              ;DURING THE ABOVE OPERATION, ONLY 'PIP
                                                                ;AND 'DPR' SHOULD BE SET
                                                                ;'MOL' & 'VV' ARE DON'T CARES

021326                          11$:

021326 104400 021334                    TYPE    ,96$            ;;TYPE ASCIZ STRING
021332 000425                           BR      95$             ;;GET OVER THE ASCIZ
                                ;;96$:  .ASCIZ  <15><12>/IF STANDBY NOT LIT - ERROR AFTER UNLOAD/
021406                          95$:
                                                                ;THIS PROVIDES A 1 SECOND "STALL"
```

```
 4545
 4546
 4547
 4548                                   ;*NOW CHANGE REGISTERS SAVED BEFORE UNLOAD COMMAND
 4549                                   ;*TO EXPECTED VALUES AFTER UNLOAD COMMAND
 4550
 4551                                   ;* - AGAIN 'MOL' $ 'VV' ARE DON'T CARES
 4552
 4553  021406  012746  020400          MOV     #PIP!DPR,-(SP)  ;SET EXPECTED FINAL RHDS1 BITS
 4554  021412  017737  160704  004760  MOV     @RHDS1,@#TMP4   ;GET PRESENT ACTUAL RHDS1 CONTENTS
 4555  021420  042737  167677  004760  BIC     #^C<MOL!VV>,@#TMP4 ;MASK OUT ALL BUT 'MOL' & 'VV'
 4556  021426  042716  010100          BIC     #MOL!VV,(SP)    ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
 4557  021432  053716  004760          BIS     @#TMP4,(SP)     ;SET EXPECTED 'MOL' & 'VV' STATES
 4558                                                         ;FROM THE ACTUAL (DON'T CARE COND.)
 4559  021436  042716  100200          BIC     #ATA!DRY,(SP)   ;CLEAR THESE ADDITIONAL RHDS1 BITS
 4560  021442  012637  004642          MOV     (SP)+,@#SAVERE+30 ;CHANGE THE SAVED RHDS1 REGISTER
 4561                                                         ;AND ADJUST THE STACK
 4562
 4563
 4564  021446  004037  042246          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4565  021452  002300                  RHCS1                   ;CHANGE RHCS1 REGISTER
 4566
 4567  021454  000001                  1                       ;1 BIT/BITS TO BE CHANGED
 4568  021456  000001                  1                       ;NEW VALUE OF GO IS 1
 4569  021460  000001                  GO                      ;CHANGE GO BIT
 4570
 4571  021462  005737  004722          TST     @#NUNIT         ;IS THERE MORE THAN ONE UNIT ?
 4572  021466  001006                  BNE     7$              ;SKIP NEXT IF MORE THAN ONE UNIT
 4573
 4574  021470  004037  042246          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4575  021474  002300                  RHCS1                   ;CHANGE RHCS1 REGISTER
 4576
 4577  021476  000001                  1                       ;1 BIT/BITS TO BE CHANGED
 4578  021500  000000                  0                       ;NEW VALUE OF SC IS 0
 4579  021502  100000                  SC                      ;CHANGE SC BIT
 4580  021504                  7$:
 4581
 4582  021504  043737  004740  004636  BIC     @#ATTENT,@#SAVERE+24    ;CLEAR APPROPIATE ATA BITS
 4583                                                                   ;FOR WORKING DRIVE IN SAVED RHAS
 4584
 4585                                   ;*NOW COMPARE REGISTERS AFTER THE UNLOAD COMMAND
 4586                                   ;*WITH EXPECTED VALUES
 4587
 4588
 4589  021512  004037  042354          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 4590                                                         ;PRESENT VALUE
 4591  021516  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 4592  021520  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
 4593  021522  000021                  17.                     ;17. REGISTERS TO BE COMPARED
 4594  021524  021530                  3$                      ;RETURN TO 3$ ON ERROR
 4595  021526  021534                  4$                      ;RETURN TO 4$ ON NO ERROR
 4596
 4597  021530  104023          3$:     ERROR   23              ;UNLOAD COMMAND GAVE
 4598  021532  000207                  RTS     PC              ;AN ERROR
```

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

                                    4$:
021534                                      TYPE    @98$            ;;TYPE ASCIZ STRING
021534  104400  021542                      BR      97$             ;;GET OVER THE ASCIZ
021540  000406                      ;;98$:  .ASCIZ  <15><12>/ON DRIVE /
                                    97$:
021556                                      MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
021556  013746  004716                      TYPDS
021562  104404                              TYPE    @100$           ;;TYPE ASCIZ STRING
021564  104400  021572                      BR      99$             ;;GET OVER THE ASCIZ
021570  000444                      ;;100$: .ASCIZ  <15><12>/IF STANDBY LIT HIT STANDBY OR START - AFTER HEAD LOAD HIT CONT/
                                    99$:
021702

021702  005037  046760                      CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR NUMBER
021706  000000                              HALT                    ;WAIT FOR CONTINUE
021710  004737  041424                      JSR     PC,@#CHECK      ;CHECKS THAT DVA,RDY,MOL & DPR = 1
021714  000240                              NOP                     ;CHECKS THAT ALL OTHER BITS = 0
021716  000240                              NOP                     ;THERE IS NO HALT IF IT FAILS - IT
021720  000240                              NOP                     ;IS USED IN THE MIDDLE OF A TEST


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


021722  013746  002460                      MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
021726  052716  000001                      BIS     #GO,(SP)        ;GET READY TO SET GO
                                                                    ;WITHOUT INTERRUPT ENABLE
021732  012677  160342                      MOV     (SP)+,@RHCS1    ;GO WITH
                                                                     ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
                                                                    ;WITH INTERRUPT DISABLED

021736  011100                              MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
021740  011305                              MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

021742  104412                              WAT                     ;WAIT FOR VV BIT TO SET
021744  002322                              RHDS1                   ;WAIT FOR RHDS1 REGISTER
021746  000100                              VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
021750  000001                              1.                      ;ALLOW 10 MICRO SECONDS
021752  000001                              1.                      ;VV MUST SET BETWEEN
                                                                    ;00 AND 20 MICRO SECONDS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 31       OFFSET AND RETURN TO CENTER LINE COMMAND
                                        ;
                                        ;*      THIS TESTS TWO COMMANDS: (1)OFFSET, (2)RETURN-TO-CENTER-LINE
                                        ;
                                        ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF)
                                        ;*      AND AN OFFSET IS GIVEN
                                        ;*      ALL REGISTERS ARE COMPARED, ONLY ATA SHOULD SET
                                        ;*      ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
                                        ;
                                        ;*      THEN A RETURN-TO-CENTER-LINE IS GIVEN
                                        ;*      ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
                                        ;*      AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)
                                        ;*      ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
                                        ;
                                        ;*      THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER
                                        ;*      VALUES OF 1 TO 377 IE. 377 TIMES
                                        ;
                                        ;;*****************************************************************
021754  000004                  T$T31:  SCOPE
021756  012737  022030  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
021764  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021770  012737  000031  004604          MOV     #31,@#TSTNM     ;SAVE TEST NUMBER
                                        ;
021776  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
022002  004737  041424                  JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
022006  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
022012  000000                          HALT                    ;STOP
022014  112737  000001  004610          MOVB    #1,@#OFSTVL     ;SET OFFSET VALUE TO 1
022022  112737  000034  004611          MOVB    #34,@#OFSTVL+1  ;SET HCI,ECI,FMT22
                                        ;
                                        ;
022030                          1$:     ;
                                        ;
022030  004737  041356                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
022034  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
022040  000240                          NOP                     ;CHECK THAT ALL OTHER BITS = 0
022042  000240                          NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
022044  000240                          NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
                                                                ;USED IN THE MIDDLE OF A SINGLE TEST
022046  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0
                                        ;
022052  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
                                        ;
                                        ;
022056  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
                                        ;
```

```
4699
4700  022062  052730  000010                BIS     #10,@(R0)+      ;LOAD 10 INTO RHCS2
4701
4702  022066  012730  001400                MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
4703
4704
4705  022072  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
4706
4707
4708  022076  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
4709
4710
4711  022102  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
4712
4713                                        ;*THE OFFSET REGISTER WILL BE INCREMENTED FROM 0 TO 377
4714
4715  022106  013730  004610                MOV     @#OFSTVL,@(R0)+ ;SET OFFSET REGISTER
4716
4717  022112  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
4718
4719  022116  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
4720
4721                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
4722
4723  022122  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4724  022124  011446                        MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
4725  022126  011246                        MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
4726                                                                ;AFTER ALL ATA BITS HAVE BEEN SET
4727  022130  013700  004742                MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
4728  022134  005012                        CLR     @R2             ;CLEAR RHCS2 AND CARRY
4729  022136  012705  000010                MOV     #8.,R5          ;COUNTER
4730  022142  006000                82$:    ROR     R0              ;GET BIT INTO CARRY
4731  022144  103002                        BCC     83$             ;BRANCH IF NO UNIT ON THIS BIT
4732  022146  012714  177777                MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
4733                                                                ;TO SET ATA BIT
4734  022152  005212                83$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
4735  022154  005305                        DEC     R5              ;COUNT
4736  022156  001371                        BNE     82$             ;BRANCH IF 8 NOT DONE
4737  022160  012612                        MOV     (SP)+,@R2       ;REINSTATE RHCS2
4738  022162  012614                        MOV     (SP)+,@R4       ;REINSTATE RHER1
4739  022164  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
4740  022166  005720                        TST     (R0)+           ;GET OVER PHAS IN R0      ——
4741
4742
4743  022170  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
4744
4745  022174  013777  002454  160076        MOV     @#OFSETC,@RHCS1 ;GET READY FOR OFSETC
4746
4747
4748                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER OFFSET
4749
4750  022202  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
4751  022206  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4752  022210  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
```

```
                                                                ;THE REGISTERS ARE SAVED
022212  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

022214  013777  004606  160044        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

022222  013746  002454            MOV     @#OFSETC,-(SP)  ;GET READY TO MOVE COMMAND
022226  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                          ;ENABLE INTERRUPT
022232  012677  160042            MOV     (SP)+,@RHCS1    ;GO WITH
                                                           ;14 IN RHCS1 FOR OFSET
                                                          ;WITH INTERRUPT ENABLED
022236  011100                    MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
022240  011305                    MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

022242  104412                    WAT                     ;WAIT FOR DRY BIT TO SET
022244  002322                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
022246  000200                    DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
022250  001750                    1000.                   ;ALLOW 10000 MICRO SECONDS
022252  000454                    300.                    ;DRY MUST SET BETWEEN
                                                          ;7000 AND 13000 MICRO SECONDS

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

022254  013746  002454            MOV     @#OFSETC,-(SP)  ;SAVE COMMAND
022260  052716  004301            BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
022264  005737  004722            TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
022270  001413                    BEQ     87$             ;BRANCH IF ONLY ONE UNIT
022272  010037  004760            MOV     R0,@#TMP4       ;GET RHCS1
022276  042737  177677  004760    BIC     #^C<IE>,@#TMP4  ;KEEP IE BIT
022304  042716  000100            BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
022310  053716  004760            BIS     @#TMP4,(SP)     ;GET IE AS IS
022314  052716  100000            BIS     #SC,(SP)        ;SET SC IN RHCS1
022320                    87$:
022320  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
022324  022600                    CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
                                                          ;AND COMMAND SHOULD BE SET
022326  001405                    BEQ     86$             ;BRANCH IF GOOD
022330  010037  001126            MOV     R0,@#$BDDAT     ;BAD DATA
022334  010137  004600            MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
022340  104021                    ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                          ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
022342  012746  030500    86$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
022346  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
022352  022605                    CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
                                                          ;SHOULD BE SET
```

```
4807  022354  001405                          BEQ     88$             ;BRANCH IF GOOD
4808  022356  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
4809  022362  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS!
4810  022366  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
4811                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
4812  022370                          88$:
4813
4814                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4815
4816
4817  022370  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4818  022374  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
4819
4820  022376  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4821  022400  000001                          1                       ;NEW VALUE OF SC IS 1
4822  022402  100000                          SC                      ;CHANGE SC BIT
4823
4824  022404  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4825  022410  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
4826
4827  022412  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4828  022414  000001                          1                       ;NEW VALUE OF ATA IS 1
4829  022416  100000                          ATA                     ;CHANGE ATA BIT
4830
4831  022420  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
4832                                                                  ;FOR WORKING DRIVE IN
4833                                                                  ;SAVED RHAS LOACTION
4834
4835                                          ;*NOW COMPARE REGISTERS AFTER AN OFSET COMMAND
4836
4837
4838  022426  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4839                                                                  ;PRESENT VALUE
4840  022432  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4841  022434  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4842  022436  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4843  022440  022444                          2$                      ;RETURN TO 2$ ON ERROR
4844  022442  022450                          3$                      ;RETURN TO 3$ ON NO ERROR
4845
4846  022444  104024                  2$:     ERROR   24              ;OFFSET COMMAND CAUSED AN ERROR
4847  022446  000207                          RTS     PC              ;GOOD DATA IS WHAT SHOULD BE THERE
4848                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
4849                                                                  ;AFTER AN OFFSET COMMAND
4850
4851  022450  013777  004740  157640  3$:     MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE ATTENTION
4852
4853
4854
4855
4856                                          ;*NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN
4857                                  ;;**************************************************************
4858
4859  022456  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
4860  022462  000240                          NOP                     ;CHECK THAT ALL OTHER BITS = 0
```

```
4861  022464  000240                          NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
4962  022466  000240                          NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
4863                                                                  ;USED IN THE MIDDLE OF A SINGLE TEST
4864  022470  013777  002456  157602          MOV     @#RETCL,@RHCS1  ;GET READY FOR RETCL
4865                                                                  ;RETURN TO CENTER LINE WITH 16 IN RHCS1
4866
4867
4868                                          ;*NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND
4869
4870  022476  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
4871  022502  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4872  022504  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
4873                                                                  ;THE REGISTERS ARE SAVED
4874  022506  000022                          18.                     ;NUMBER OF REGISTERS
4875                                                                  ;SAVED = 18.
4876
4877  022510  013777  004606  157550          MOV     @#RP4VEC,@RP\EC ;SET RP04 VECTOR ADDRESS
4878                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
4879                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4880                                                                  ;'TIME' WILL ONLY SAVE
4881                                                                  ;CURRENT CYLINDER ADDRESS
4882                                                                  ;AND LOOK AHEAD REGISTERS
4883
4884
4885  022516  013746  002456                  MOV     @#RETCL,-(SP)   ;GET READY TO MOVE COMMAND
4886  022522  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
4887                                                                  ;ENABLE INTERRUPT
4888  022526  012677  157546                  MOV     (SP)+,@RHCS1    ;GO WITH
4889                                                                  ;16 IN RHCS1 FOR RETURN TO CENTER LINE
4890                                                                  ;WITH INTERRUPT ENABLED
4891  022532  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
4892  022534  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
4893
4894
4895  022536  104412                          WAT                     ;WAIT FOR DRY BIT TO SET
4896  022540  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
4897  022542  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4898  022544  001750                          1000.                   ;ALLOW 10000 MICRO SECONDS
4899  022546  001750                          1000.                   ;DRY MUST SET BETWEEN
4900                                                                  ;00 AND 20000 MICRO SECONDS
4901
4902                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4903                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
4904
4905  022550  013746  002456                  MOV     @#RETCL,-(SP)   ;SAVE COMMAND
4906  022554  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
4907  022560  005737  004722                  TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
4908  022564  001413                          BEQ     90$             ;BRANCH IF ONLY ONE UNIT
4909  022566  010037  004760                  MOV     R0,@#TMP4       ;GET RHCS1
4910  022572  042737  177677  004760          BIC     #^CIE,@#TMP4    ;KEEP IE BIT
4911  022600  042716  000100                  BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
4912  022604  053716  004760                  BIS     @#TMP4,(SP)     ;GET IE AS IS
4913  022610  052716  100000                  BIS     #SC,(SP)        ;SET SC IN RHCS1
4914  022614                          90$:
```

```
4915  022614  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4916  022620  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
4917                                                                  ;AND COMMAND SHOULD BE SET
4918  022622  001405                          BEQ     89$             ;BRANCH IF GOOD
4919  022624  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
4920  022630  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
4921  022634  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
4922                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4923  022636  012746  030500          89$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
4924  022642  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4925  022646  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
4926                                                                  ;SHOULD BE SET
4927  022650  001405                          BEQ     91$             ;BRANCH IF GOOD
4928  022652  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
4929  022656  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
4930  022662  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
4931                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
4932  022664                          91$:
4933
4934                                          ;*NOW CHANGE SAVED REGISTER TO EXPECTED VALUE
4935
4936
4937  022664  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4938  022670  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
4939
4940  022672  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4941  022674  000001                          1                       ;NEW VALUE OF SC IS 1
4942  022676  100000                          SC                      ;CHANGE SC BIT
4943
4944  022700  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4945  022704  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
4946
4947  022706  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4948  022710  000001                          1                       ;NEW VALUE OF ATA IS 1
4949  022712  100000                          ATA                     ;CHANGE ATA BIT
4950
4951  022714  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
4952                                                                  ;FOR WORKING DRIVE IN
4953                                                                  ;SAVED RHAS LOACTION
4954  022722  004037  041270                  JSR     R0,@#FILLRE     ;MOV BIT15!HCI!ECI!FMT22 INTO SAVED RHOF
4955  022726  002310                          RHOF                    ;SAVED REGISTER TO CHANGE
4956  022730  116000                          BIT15!HCI!ECI!FMT22                     ;DATA
4957
4958                                          ;*NOW COMPARE REGISTERS AFTER RETURN-TO-CENTER-LINE
4959
4960  022732  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4961                                                                  ;PRESENT VALUE
4962  022736  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4963  022740  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4964  022742  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4965  022744  022750                          4$                      ;RETURN TO 4$ ON ERROR
4966  022746  022754                          5$                      ;RETURN TO 5$ ON NO ERROR
4967  022750  104025          4$:             ERROR   25              ;RETURN TO CENTER-LINE
4968  022752  000207                          RTS     PC              ;COMMAND CAUSED AN ERROR
```

```
                                                                        ;GOOD DATA HAS WHAT SHOULD
                                                                        ;BE THERE
                                                                        ;RECEIVED DATA HAS WHAT WAS
                                                                        ;THERE AFTER COMMAND

022754                                  5$:

022754  004737  041366                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
022760  105237  004610                        INCB    @#OFSTVL        ;GET NEXT OFFSET VALUE
022764  132737  000100  004610                BITB    #100,@#OFSTVL   ;SEE IF UNUSED BIT 6  IS ON
022772  001403                                BEQ     7$              ;NO SO DO SOME MORE
022774  062737  000100  004610                ADD     #100,@#OFSTVL   ;YES SO BY-PASS IT
023002  105737  004610              7$:       TSTB    @#OFSTVL        ;IF ZERO ALL COMBINATIONS ARE
                                                                        ;COMPLETE
023006  001001                                BNE     6$              ;BRANCH IF 377 NOT DONE
023010  000402                                BR      TST32   :       BRANCH TO NEXT TEST
023012  000137  022030              6$:       JMP     @#1$            ;JUMP BECAUSE 377 NOT DONE
```

```
                                ;;**********************************************************************
                                ;*TEST 32       OFFSET COMMAND
                                ;
                                ;*      THIS TEST WILL ONLY GIVE REPEATED OFFSETS
                                ;
                                ;;**********************************************************************
023016  000004                  TST32:  SCOPE
023020  012737  000764  004756          MOV     #500.,@#TMP1    ;COUNTER
023026                          1$:
023026  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023032  012737  000032  004604          MOV     #32.,@#TSTNM    ;SAVE TEST NUMBER
                
023040  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
023044  004737  041446                  JSR     PC,@#CHECK      ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
023050  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
023054  000000                          HALT                    ;STOP
023056  004037  041352                  JSR     R0,@#OFSET      ;OFSET
023062  000260                          260                     ;260 IN OFSET REGISTER
                                                                ;OFSET -1200 MICRO INCHES

023064  013746  002454                  MOV     @#OFSETC,-(SP)  ;GET READY TO MOVE COMMAND
023070  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
023074  012677  157200                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;14 IN RHCS1 FOR OFSET
                                                                ;WITH INTERRUPT ENABLED

023100  104412                          WAT                     ;WAIT FOR DRY BIT TO SET
023102  002322                          RHCS1                   ;WAIT FOR RHDS1 REGISTER
023104  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
023106  000536                          350.                    ;ALLOW 3500 MICRO SECONDS
023110  000536                          350.                    ;DRY MUST SET BETWEEN
                                                                ;00 AND 7000 MICRO SECONDS
023112  032777  040000  157202          BIT     #ERR,@RHDS1     ;IS ERR SET?
023120  001417                          BEQ     2$              ;NO
023122  104026                          ERROR   26              ;REPEATED OFFSETS CAUSED AN ERROR
023124  000004                          SCOPE

023126  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

023132  013746  002426                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
023136  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
023142  012677  157132                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;6 IN RHCS1 FOR RECALIBRATE
```

```
5044                                                    ;WITH INTERRUPT ENABLED
5045
5046  023146  104412                        WAT         ;WAIT FOR DRY BIT TO SET
5047  023150  002322                        RHDS1       ;WAIT FOR RHDS1 REGISTER
5048  023152  000200                        DRY         ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5049  023154  060650                        25000.      ;ALLOW 250000 MICRO SECONDS
5050  023156  060650                        25000.      ;DRY MUST SET BETWEEN
5051                                                    ;00 AND 500000 MICRO SECONDS
5052  023160  005337  004756        2$:     DEC     @#TMP1      ;COUNT DOWN
5053  023164  001401                BEQ     .+TST33     ;BRANCH IF DONE
5054  023166  000717                BR      1$          ;GO BACK AND DO IT AGIEN
5055
```

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

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

                                ;;*****************************************************************
                                ;;*****************************************************************
023170  000004                  ↑ST33:  SCOPE
023172  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023176  012737  000033  004634          MOV     #33,@#↑STNM     ;SAVE TEST NUMBER

023204  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*FILL WRITE FROM BUFFER WITH HEADER

023210  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
023214  002470                          WRFROM                  ;LOCATION WHERE SAVED
023216  000004                          4                       ;NUMBER OF WORDS SAVED
023220  010000                          10000                   ;FIRST DATA WORD
023222  000000                          0                       ;SECOND DATA WORD
023224  000000                          0                       ;THIRD DATA WORD
023226  000000                          0                       ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH DATA

023230  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
023234  002500                          WRFROM+10               ;STARTING FROM WRFROM+10
023236  000400                          256.                    ;256. WORDS
023240  000000                          0                       ;FILL WITH 0
```

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

023242  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
023246  003534                          REINTO                  ;LOCATION WHERE SAVED
023250  000004                          4                       ;NUMBER OF WORDS SAVED
023252  010000                          10000                   ;FIRST DATA WORD
023254  000000                          0                       ;SECOND DATA WORD
023256  000000                          0                       ;THIRD DATA WORD
023260  000000                          0                       ;FOURTH DATA WORD
023262  004037  041236                  JSP     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
023266  003544                          REINTO+10                       ;STARTING FROM REINTO+10
023270  000400                          256.                    ;256. WORDS
023272  000000                          0                       ;FILL WITH 0

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

023274  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
023300  000000                          0                       ;CYLINDER 0
023302     000                  .BYTE   0                       ;SECTOR 0
023303     000                  .BYTE   0                       ;TRACK 0
023304  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                                ;4 HEADER WORDS
023306  002470                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
023310  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023312  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
023314  002444                          WRIFOR                  ;GET READY TO DO A WRIFOR
                                                ;WRITE HEADER AND DATA WITH 62 IN RHCS1

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

023316  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
023322  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
023324  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
023326  000021                          17.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 17.

023330  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
023334  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
023340  000000                          HALT                    ;STOP
```

```
023342  013777  004636  156716          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

023350  013746  002444                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
023354  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
023360  012677  156714                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED
023364  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
023366  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

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

023370  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
023372  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
023374  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
023376  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
023400  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

023402  013746  002444                  MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
023406  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
023412  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023416  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
023420  001405                          BEQ     64$             ;BRANCH IF GOOD
023422  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
023426  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
023432  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
023434  012746  010530          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
023440  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023444  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
023446  001405                          BEQ     66$             ;BRANCH IF GOOD
023450  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
023454  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
023460  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
023462                          66$:

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

```
5218  023462  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
5219  023466  002272                  RHWC                    ;SAVED REGISTER TO CHANGE
5220  023470  000000                  0                       ;DATA
5221  023472  004037  041270          JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2> INTO SAVED RHBA
5222  023476  002274                  RHBA                    ;SAVED REGISTER TO CHANGE
5223  023500  003500                  WRFROM+<260.*2>                 ;DATA
5224  023502  004037  041270          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5225  023506  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
5226  023510  000001                  1                       ;DATA
5227
5228                                  ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
5229                                  ;*WITH REGISTERS AFTER COMMAND
5230
5231
5232  023512  004037  042354          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5233                                                          ;PRESENT VALUE
5234  023516  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5235  023520  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
5236  023522  000021                  17.                     ;17. REGISTERS TO BE COMPARED
5237  023524  023530                  1$                      ;RETURN TO 1$ ON ERROR
5238  023526  023534                  2$                      ;RETURN TO 2$ ON NO ERROR
5239
5240  023530  104027          1$:     ERROR   27              ;WRITE HEADER AND DATA
5241  023532  000207                  RTS     PC              ;CAUSED IMPROPER REGISTER
5242                                                          ;CHANGE
5243                                                          ;GOOD DATA GIVES WHAT SHOULD
5244                                                          ;BE THERE
5245                                                          ;RECEIVED DATA GIVES WHAT
5246                                                          ;WAS THERE AFTER COMMAND
5247
5248                                  ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
5249                                  ;*NOTHING GOT CHANGED
5250
5251  023534          2$:
5252
5253  023534  004037  043404          JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5254  023540  003534                  REINTO                  ;GOOD DATA STARTS FROM REINTO
5255  023542  002470                  WRFROM                  ;TEST DATA STARTS FROM WRFROM
5256  023544  000404                  260.                    ;260. WORDS TO BE COMPARED
5257  023546  023552                  3$                      ;RETURN TO 3$ ON ERROR
5258  023550  023556                  4$                      ;RETURN TO 4$ ON NO ERROR
5259
5260
5261  023552  104030          3$:     ERROR   30              ;WRITE HEADER AND DATA
5262  023554  000207                  RTS     PC              ;CHANGED WRITE FROM BUFFER
5263
5264                                  ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
5265                                  ;*READ INTO BUFFER IS FILLED WITH ONES
5266
5267  023556          4$:
5268
5269  023556  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5270                                                          ;R3-RHDS1, R4-RHER1
5271                                                          ;GIVE RH-11 INITIALIZE
```

```
                                                        ;SETUP UNIT NUMBER
023562  004037  041236                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
023566  003534                      REINTO                  ;STARTING FROM REINTO
023570  000404                      260.                    ;260. WORDS
023572  177777                      -1                      ;FILL WITH -1


                                    ;*NOW FILL COMMAND


023574  004037  043340              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
023600  000000                      0                       ;CYLINDER 0
023602     000              .BYTE   0                       ;SECTOR 0
023603     000              .BYTE   0                       ;TRACK 0
023604  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                            ;4 HEADER WORDS
023606  003534                      REINTO                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = REINTO
023610  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023612  014000                      ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                            ;INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
023614  002450                      REFOR                   ;GET READY TO DO A REFOR
                                                            ;READ HEADER AND DATA WITH 72 IN RHCS1


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

023616  004037  041534              JSR     R0,@#SAVER      ;SAVE REGISTERS
023622  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
023624  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
                                                            ;THE REGISTERS ARE SAVED
023626  000022                      18.                     ;NUMBER OF REGISTERS
                                                            ;SAVED = 18.

023630  004737  041446              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                            ;AND THAT NO STATUS BITS ARE STUCK = 1
023634  104400  066402              TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                            ;THE FIRST SET OF BITS DON'T = 1
023640  000000                      HALT                    ;STOP

023642  013777  004606  156416      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


023650  013746  002450              MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
023654  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                            ;ENABLE INTERRUPT
023660  012677  156414              MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;72 IN RHCS1 FOR READ DATA
                                                    ;WITH INTERRUPT ENABLED
023664  011100                      MOV     @R1,R0      ;SAVE RHCS1 DURING ABOVE OPERATION
023666  011305                      MOV     @R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION

023670  104412                      WAT                 ;WAIT FOR RDY BIT TO SET
023672  002300                      RHCS1               ;WAIT FOR RHCS1 REGISTER
023674  0002C0                      RDY                 ;WAIT FOR RDY BIT IN RHCS1 REGISTER
023676  001614                      908.                ;ALLOW 9080 MICRO SECONDS
023700  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

023702  013746  002450              MOV     @#REFOR,-(SP)   ;SAVE COMMAND
023706  052716  004101              BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
023712  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023716  022600                      CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                        ;AND COMMAND SHOULD BE SET
023720  001405                      BEQ     67$             ;BRANCH IF GOOD
023722  010037  001126              MOV     R0,@#$BDDAT     ;BAD DATA
023726  010137  004600              MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
023732  104021                      ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                        ;COMMAND AND IE!GO!DVA SHOULD BE SET
023734  012746  010500      67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
023740  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023744  022605                      CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
023746  001405                      BEQ     69$             ;BRANCH IF GOOD
023750  010537  001126              MOV     R5,@#$BDDAT     ;BAD DATA
023754  010337  004600              MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
023760  104063                      ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
023762                      69$:

                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

023762  004037  041270              JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
023766  002272                      RHWC                    ;SAVED REGISTER TO CHANGE
023770  000000                      0                       ;DATA
023772  004037  041270              JSR     R0,@#FILLRE     ;MOV REINTO+(260.*2) INTO SAVED RHBA
023776  002274                      RHBA                    ;SAVED REGISTER TO CHANGE
024000  004544                      REINTO+(260.*2)             ;DATA
024002  004037  041270              JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
024006  002304                      RHDST                   ;SAVED REGISTER TO CHANGE
024010  000001                      1                       ;DATA

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


024012  004037  042354              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
5380
5381  024016  004612                        SAVERE                  ;PRESENT VALUE
5382  024020  002354                        WC                      ;GOOD DATA SAVED IN 'SAVERE'
5383  024022  000022                        18.                     ;TEST DATA STARTING FROM 'RHWC'
5384  024024  024030                        5$                      ;18. REGISTERS TO BE COMPARED
5385  024026  024034                        6$                      ;RETURN TO 5$ ON ERROR
5386                                                                ;RETURN TO 6$ ON NO ERROR
5387  024030  104031              5$:       ERROR   31              ;READ HEADER AND DATA CAUSED
5388  024032  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
5389                                                                ;GOOD DATA GIVES WHAT SHOULD
5390                                                                ;BE THERE
5391                                                                ;RECEIVED DATA GIVES WHAT WAS
5392                                                                ;THERE AFTER COMMAND
5393
5394                                        ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
5395                                        ;*THE READ WAS GOOD
5396
5397  024034              6$:
5398
5399  024034  004037  043404                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5400  024040  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
5401  024042  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
5402  024044  000404                        260.                    ;260. WORDS TO BE COMPARED
5403  024046  024052                        7$                      ;RETURN TO 7$ ON ERROR
5404  024050  024056                        10$                     ;RETURN TO 10$ ON NO ERROR
5405
5406
5407  024052  104032              7$:       ERROR   32              ;WRITE HEADER AND DATA
5408  024054  000207                        RTS     PC              ;FOLLOWED BY A READ HEADER
5409                                                                ;AND DATA GAVE A READ ERROR
5410                                                                ;ERROR MAY BE IN READ OR WRITE
5411  024056              10$:
5412
```

```
                                ;;*****************************************************************
                                ;*TEST 34        READ DATA (0'S)
                                ;
                                ;*      THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST
                                ;*      THE WRITE FROM BUFFER IS FILLED WITH ALL 0S
                                ;*      THE COMMAND IS FILLED, THEN ALL REGISTERS SAVED
                                ;*      THEN READ DATA COMMAND IS GIVEN
                                ;*      ALL REGISTERS ARE COMPARED FOR PROPER VALUES
                                ;*      READ DATA INTO 'READ INTO' BUFFER IS CHECKED
                                ;
                                ;;*****************************************************************
024056  000004                TST34:  SCOPE
024060  012706  001000                MOV     #STACK,SP       ;RESET STACK
024064  012737  000034  004604        MOV     #34,@#^TSTNM    ;SAVE TEST NUMBER
                                ;
024072  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER
                                ;
                                ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
                                ;
024076  004037  041236                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
024102  002470                        WRFROM                  ;STARTING FROM WRFROM
024104  000400                        256.                    ;256. WORDS
024106  000000                        0                       ;FILL WITH 0
                                ;
                                ;*NOW THE READ DATA COMMAND WILL BE FILLED
                                ;
                                ;
024110  004037  043340                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
024114  000000                        0                       ;CYLINDER 0
024116     000                .BYTE   0                       ;SECTOR 0
024117     000                .BYTE   0                       ;TRACK 0
024120  177400                        -256.                   ;WORD COUNT = 256.
024122  003534                        REINTO                  ;BUS ADDRESS
                                                              ;STARTING ADDRESS OF DATA
                                                              ;BUFFER = REINTO
024124  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
024126  014000                        ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                              ;INHIBIT ECC CORRECTION
                                                              ;DO NOT INHIBIT HEADER COMPARE
024130  002446                        READAT                  ;GET READY TO DO A READAT
                                                      ;READ DATA WITH 70 IN RHCS1
                                ;
                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
                                ;
024132  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
024136  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
024140  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                              ;THE REGISTERS ARE SAVED
```

```
5467  024142  000022                          18.                     ;NUMBER OF REGISTERS
5468                                                                  ;SAVED = 18.
5469
5470  024144  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5471                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
5472  024150  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
5473                                                                  ;THE FIRST SET OF BITS DON'T = 1
5474  024154  000000                          HALT                    ;STOP
5475
5476  024156  013777  004606  156102          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5477                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5478                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5479                                                                  ;'TIME' WILL ONLY SAVE
5480                                                                  ;CURRENT CYLINDER ADDRESS
5481                                                                  ;AND LOOK AHEAD REGISTERS
5482
5483
5484  024164  013746  002446                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
5485  024170  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5486                                                                  ;ENABLE INTERRUPT
5487  024174  012677  156100                  MOV     (SP)+,@RHCS1    ;GO WITH
5488                                                                   ;70 IN RHCS1 FOR READ DATA
5489                                                                  ;WITH INTERRUPT ENABLED
5490  024200  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5491  024202  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5492
5493
5494  024204  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
5495  024206  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
5496  024210  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5497  024212  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
5498  024214  001507                          839.                    ;RDY MUST SET BETWEEN
5499                                                                  ;690 AND 17470 MICRO SECONDS
5500
5501                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5502                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
5503
5504  024216  013746  002446                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
5505  024222  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
5506  024226  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5507  024232  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5508                                                                  ;AND COMMAND SHOULD BE SET
5509  024234  001405                          BEQ     64$             ;BRANCH IF GOOD
5510  024236  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
5511  024242  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
5512  024246  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
5513                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
5514  024250  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
5515  024254  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5516  024260  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5517                                                                  ;SHOULD BE SET
5518  024262  001405                          BEQ     66$             ;BRANCH IF GOOD
5519  024264  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
5520  024270  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
```

```
5521  024274  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
5522                                                                  ;MOL!DPR!VV SHOULD BE SET
5523  024276                          66$:
5524
5525                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5526
5527  024276  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
5528  024302  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
5529  024304  000000                          0                       ;DATA
5530  024306  004037  041270                  JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
5531  024312  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
5532  024314  004534                          REINTO+<256.*2>             ;DATA
5533  024316  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5534  024322  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
5535  024324  000001                          1                       ;DATA
5536
5537                                          ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
5538                                          ;*AFTER COMMAND
5539
5540
5541  024326  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5542                                                                  ;PRESENT VALUE
5543  024332  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5544  024334  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5545  024336  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5546  024340  024344                          1$                      ;RETURN TO 1$ ON ERROR
5547  024342  024350                          2$                      ;RETURN TO 2$ ON NO ERROR
5548
5549  024344  104033                  1$:     ERROR   33              ;READ DATA CAUSED IMPROPER REGISTER
5550  024346  000207                          RTS     PC              ;CHANGE
5551                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
5552                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
5553                                          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
5554                                          ;*WAS GOOD
5555
5556  024350                          2$:
5557
5558  024350  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5559  024354  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
5560  024356  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
5561  024360  000400                          256.                    ;256. WORDS TO BE COMPARED
5562  024362  024366                          3$                      ;RETURN TO 3$ ON ERROR
5563  024364  024372                          4$                      ;RETURN TO 4$ ON NO ERROR
5564
5565
5566  024366  104034                  3$:     ERROR   34              ;READ DATA COMMAND
5567  024370  000207                          RTS     PC              ;READ INCORRECTLY
5568
5569  024372                          4$:
```

```
                        ;;*****************************************************************
                        ;*TEST 35       WRITE/READ DATA (1'S & 125252)

                        ;*      THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0
                        ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES
                        ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                        ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                        ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                        ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                        ;*      200 ONES AND 56 125252
                        ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                        ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
                        ;*
                        ;*
                        ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                        ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                        ;*
                        ;*      THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS
                        ;*      AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES
                        ;*      AND 56 OF 377.
                        ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                        ;*      GO IS GIVEN
                        ;*
                        ;*
                        ;*      ALL REGISTERS ARE CHECKED
                        ;*      READ DATA IS CHECKED

                        ;;*****************************************************************
024372  000004         ↑ST35:  SCOPE
024374  012706  001000         MOV     #STACK,SP       ;RESET STACK
024400  012737  000035  004604 MOV     #35.,@#↑STNM    ;SAVE TEST NUMBER

024406  004737  041366         JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                       ;R3-RHDS1, R4-RHER1
                                                       ;GIVE RH-11 INITIALIZE
                                                       ;SETUP UNIT NUMBER

                        ;*NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252

024412  004037  041236         JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
024416  002470                 WRFROM                  ;STARTING FROM WRFROM
024420  000310                 200.                    ;200. WORDS
024422  177777                 -1                      ;FILL WITH -1

024424  004037  041236         JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+(200.*2)
024430  003310                 WRFROM+(200.*2)         ;STARTING FROM WRFROM+(200.*2)
024432  000070                 56.                     ;56. WORDS
024434  125252                 125252                  ;FILL WITH 125252


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

```
5624
5625  024436  004037  041236                JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
5626  024442  003534                        REINTO                  ;STARTING FROM REINTO
5627  024444  000310                        200.                    ;200. WORDS
5628  024446  177777                        -1                      ;FILL WITH -1
5629
5630  024450  004037  041236                JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
5631  024454  004354                        REINTO+(200.*2)                 ;STARTING FROM REINTO+(200.*2)
5632  024456  000070                        56.                     ;56. WORDS
5633  024460  125252                        125252                  ;FILL WITH 125252
5634
5635
5636                                        ;*NOW WRITE DATA COMMAND WILL BE LOADED
5637
5638
5639  024462  004037  043340                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5640  024466  000000                        0                       ;CYLINDER 0
5641  024470     000                .BYTE   0                       ;SECTOR 0
5642  024471     000                .BYTE   0                       ;TRACK 0
5643  024472  177470                        -200.                   ;WORD COUNT = 200.
5644  024474  002470                        WRFROM                  ;BUS ADDRESS
5645                                                                ;STARTING ADDRESS OF DATA
5646                                                                ;BUFFER = WRFROM
5647  024476  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5648  024500  010000                        FMT22                   ;16 BITS PER WORD FORMAT
5649                                                                ;DO NOT INHIBIT ECC CORRECTION
5650                                                                ;DO NOT INHIBIT HEADER COMPARE
5651  024502  002442                        WRIDAT                  ;GET READY TO DO A WRIDAT
5652                                                  ;WRITE DATA WITH 60 IN RHCS1
5653
5654
5655                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
5656
5657  024504  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
5658  024510  002272                        R4WC                    ;R4WC IS THE FIRST REGISTER SAVED
5659  024512  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
5660                                                                ;THE REGISTERS ARE SAVED
5661  024514  000022                        18.                     ;NUMBER OF REGISTERS
5662                                                                ;SAVED = 18.
5663
5664  024516  004737  041446                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5665                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
5666  024522  104400  066402                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
5667                                                                ;THE FIRST SET OF BITS DON'T = 1
5668  024526  000000                        HALT                    ;STOP
5669
5670  024530  013777  004606  155530        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5671                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
5672                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5673                                                                ;'TIME' WILL ONLY SAVE
5674                                                                ;CURRENT CYLINDER ADDRESS
5675                                                                ;AND LOOK AHEAD REGISTERS
5676
5677
```

```
5678  024536  013746  002442                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
5679  024542  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5680                                                                  ;ENABLE INTERRUPT
5681  024546  012677  155526                MOV     (SP)+,@RHCS1    ;GO WITH
5682                                                                   ;GO IN RHCS1 FOR WRITE DATA
5683                                                                  ;WITH INTERRUPT ENABLED
5684  024552  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5685  024554  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5686
5687                                        ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
5688
5689
5690  024556  104412                        WAT                     ;WAIT FOR RDY BIT TO SET
5691  024560  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
5692  024562  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5693  024564  001614                        900.                    ;ALLOW 9080 MICRO SECONDS
5694  024566  001507                        839.                    ;RDY MUST SET BETWEEN
5695                                                                  ;690 AND 17470 MICRO SECONDS
5696
5697                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5698                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
5699
5700  024570  013746  002442                MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
5701  024574  052716  000401                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
5702  024600  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5703  024604  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5704                                                                  ;AND COMMAND SHOULD BE SET
5705  024606  001405                        BEQ     64$             ;BRANCH IF GOOD
5706  024610  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
5707  024614  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
5708  024620  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
5709                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
5710  024622  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
5711  024626  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5712  024632  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5713                                                                  ;SHOULD BE SET
5714  024634  001405                        BEQ     66$             ;BRANCH IF GOOD
5715  024636  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
5716  024642  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
5717  024646  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
5718                                                                  ;MOL!DPR!VV SHOULD BE SET
5719  024650                        66$:
5720
5721                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
5722
5723  024650  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
5724  024654  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
5725  024656  000000                        0                       ;DATA
5726  024660  004037  041270                JSR     R0,@#FILLRE     ;MOV WRFROM+<200.*2> INTO SAVED RHBA
5727  024664  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
5728  024666  003310                        WRFROM+<200.*2>                 ;DATA
5729  024670  004037  041270                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
5730  024674  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
5731  024676  000001                        1                       ;DATA
```

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

024700  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
024704  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
024706  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
024710  000022                          18.                     ;18. REGISTERS TO BE COMPARED
024712  024716                          1$                      ;RETURN TO 1$ ON ERROR
024714  024722                          2$                      ;RETURN TO 2$ ON NO ERROR

024716  104035          1$:             ERROR   35              ;WRITE DATA COMMAND CAUSED
024720  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER COMMAND

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

024722          2$:

024722  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
024726  003534                          REINTO                  ;GOOD DATA STARTS FROM REINTO
024730  002470                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
024732  000400                          256.                    ;256. WORDS TO BE COMPARED
024734  024740                          3$                      ;RETURN TO 3$ ON ERROR
024736  024744                          4$                      ;RETURN TO 4$ ON NO ERROR

024740  104036          3$:             ERROR   36              ;WRITE DATA COMMAND CHANGED
024742  000207                          RTS     PC              ;WRITE FROM BUFFER

                                        ;*NOW A READ DATA COMMAND WILL BE GIVEN

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

024744          4$:

024744  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
024750  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
024754  003534                          REINTO                  ;STARTING FROM REINTO
024756  000310                          200.                    ;200. WORDS
024760  000000                          0                       ;FILL WITH 0

024762  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
024766  004354                          REINTO+(200.*2)                 ;STARTING FROM REINTO+(200.*2)
024770  000070                          56.                     ;56. WORDS
024772  000377                          377                     ;FILL WITH 377
```

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

025774  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
025000  002470                          WRFROM                  ;STARTING FROM WRFROM
025002  000310                          200.                    ;200. WORDS
025004  177777                          -1                      ;FILL WITH -1

025006  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
025012  003310                          WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
025014  000070                          56.                     ;56. WORDS
025016  000377                          377                     ;FILL WITH 377


                                        ;*NOW FILL COMMAND


025020  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
025024  000000                          0                       ;CYLINDER 0
025026     000                  .BYTE   0                       ;SECTOR 0
025027     000                  .BYTE   0                       ;TRACK 0
025030  177470                          -200.                   ;WORD COUNT = 200.
025032  003534                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
025034  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
025036  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
025040  002446                          READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS!


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

025042  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
025046  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
025050  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
025052  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

025054  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
025060  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
025064  000000                          HALT                    ;STOP

025066  013777  004606  155172          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
```

```
5840                                                                    ;AND LOOk AHEAD REGISTERS
5841
5842
5843  025074  013746  002446          MOV     @#READAT,-(SP)    ;GET READY TO MOVE COMMAND
5844  025100  052716  000101          BIS     #GO!IE,(SP)       ;GET READY TO SET 'GO' AND
5845                                                            ;ENABLE INTERRUPT
5846  025104  012677  155170          MOV     (SP)+,@RHCS1      ;GO WITH
5847                                                             ;70 IN RHCS1 FOR READ DATA
5848                                                            ;WITH INTERRUPT ENABLED
5849  025110  011100                  MOV     @R1,R0            ;SAVE RHCS1 DURING ABOVE OPERATION
5850  025112  011305                  MOV     @R3,R5            ;SAVE RHDS1 DURING ABOVE OPERATION
5851
5852
5853  025114  104412                  WAT                       ;WAIT FOR RDY BIT TO SET
5854  025116  002300                  RHCS1                     ;WAIT FOR RHCS1 REGISTER
5855  025120  000200                  RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5856  025122  001614                  908.                      ;ALLOW 9080 MICRO SECONDS
5857  025124  001507                  839.                      ;RDY MUST SET BETWEEN
5858                                                            ;690 AND 17470 MICRO SECONDS
5859
5860                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5861                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
5862
5863  025126  013746  002446          MOV     @#READAT,-(SP)    ;SAVE COMMAND
5864  025132  052716  004101          BIS     #IE!GO!DVA,(SP)   ;INCLUDE IE!GO!DVA
5865  025136  011637  001124          MOV     (SP),@#$GDDAT     ;SAVE FOR PRINTOUT
5866  025142  022600                  CMP     (SP)+,R0          ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5867                                                            ;AND COMMAND SHOULD BE SET
5868  025144  001405                  BEQ     67$               ;BRANCH IF GOOD
5869  025146  010037  001126          MOV     R0,@#$BDDAT       ;BAD DATA
5870  025152  010137  004600          MOV     R1,@#REGADR       ;FAILING REGISTER RHCS1
5871  025156  104021                  ERROR   21                ;DURING ABOVE OPERATION ONLY
5872                                                            ;COMMAND AND IE!GO!DVA SHOULD BE SET
5873  025160  012746  010500  67$:    MOV     #MOL!DPR!VV,-(SP)         ;SAVE BITS SET DURING OPERATION IN RHCS1
5874  025164  011637  001124          MOV     (SP),@#$GDDAT     ;SAVE FOR PRINTOUT
5875  025170  022605                  CMP     (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5876                                                            ;SHOULD BE SET
5877  025172  001405                  BEQ     69$               ;BRANCH IF GOOD
5878  025174  010537  001126          MOV     R5,@#$BDDAT       ;BAD DATA
5879  025200  010337  004600          MOV     R3,@#REGADR       ;FAILING REGISTER RHDS1
5880  025204  104063                  ERROR   63                ;DURING ABOVE OPERATION ONLY
5881                                                            ;MOL!DPR!VV SHOULD BE SET
5882  025206                  69$:
5883
5884                                  ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5885
5886  025206  004037  041270          JSR     R0,@#FILLRE       ;MOV 0 INTO SAVED RHWC
5887  025212  002272                  RHWC                      ;SAVED REGISTER TO CHANGE
5888  025214  000000                  0                         ;DATA
5889  025216  004037  041270          JSR     R0,@#FILLRE       ;MOV REINTO+<200.*2> INTO SAVED RHBA
5890  025222  002274                  RHBA                      ;SAVED REGISTER TO CHANGE
5891  025224  004354                  REINTO+<200.*2>                   ;DATA
5892  025226  004037  041270          JSR     R0,@#FILLRE       ;MOV 1 INTO SAVED RHDST
5893  025232  002304                  RHDST                     ;SAVED REGISTER TO CHANGE
```

```
5894  025234  000001                          1                       ;DATA
5895
5896                                          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
5897                                          ;*WITH REGISTERS AFTER COMMAND
5898
5899
5900  025236  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5901                                                                    ;PRESENT VALUE
5902  025242  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5903  025244  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5904  025246  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5905  025250  025254                          5$                      ;RETURN TO 5$ ON ERROR
5906  025252  025260                          6$                      ;RETURN TO 6$ ON NO ERROR
5907
5908  025254  104033                  5$:     ERROR   33              ;READ DATA CAUSED IMPROPER
5909  025256  000207                          RTS     PC              ;REGISTER CHANGE
5910                                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
5911                                                                    ;RECEIVED DATA GIVES WHAT WAS THERE
5912                                                                    ;AFTER COMMAND
5913
5914                                          ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
5915
5916  025260                          6$:
5917
5918  025260  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5919  025264  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
5920  025266  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
5921  025270  000400                          256.                    ;256. WORDS TO BE COMPARED
5922  025272  025276                          7$                      ;RETURN TO 7$ ON ERROR
5923  025274  025302                          10$                     ;RETURN TO 10$ ON NO ERROR
5924
5925
5926  025276  104034                  7$:     ERROR   34              ;INCORRECT DATA AFTER
5927  025300  000207                          RTS     PC              ;WRITE DATA FOLLOWED BY A
5928                                                                    ;READ DATA
5929  025302                          10$:
5930
5931
5932
```

```
                                        ;;*****************************************************************
                                        ;*TEST 36       WRITE/READ DATA (125252)
                                        ;*
                                        ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                        ;*      TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252
                                        ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                        ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                        ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                                        ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                        ;*      256 OF 125252
                                        ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                        ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATI
                                        ;*
                                        ;*
                                        ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                        ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                                        ;*
                                        ;*      THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS
                                        ;*      WRITE FROM BUFFER IS FILLED WITH
                                        ;*      256 OF 125252
                                        ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                        ;*      GO IS GIVEN
                                        ;*
                                        ;*
                                        ;*      ALL REGISTERS ARE CHECKED
                                        ;*      READ DATA IS CHECKED
                                        ;;*****************************************************************
025302  000004                  TST36:  SCOPE
025304  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025310  012737  000036  004604          MOV     #36,@#TSTNM     ;SAVE TEST NUMBER
025316  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                        ;*NOW FILL WRITE FROM BUFFER - 256 OF 125252
025322  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
025326  002470                          WRFROM                  ;STARTING FROM WRFROM
025330  000400                          256.                    ;256. WORDS
025332  125252                          125252                  ;FILL WITH 125252
                                        ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
                                        ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                        ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
025334  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
025340  003534                          REINTO                  ;STARTING FROM REINTO
025342  000400                          256.                    ;256. WORDS
025344  125252                          125252                  ;FILL WITH 125252
```

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

025246  004037  043340                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
025352  000000                        0                       ;CYLINDER 0
025354     000              .BYTE   0                       ;SECTOR 0
025355     000              .BYTE   0                       ;TRACK 0
025356  177400                        -256.                   ;WORD COUNT = 256.
025360  002470                        WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
025362  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
025364  010000                        FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
025366  002442                        WRIDAT                  ;GET READY TO DO A WRIDAT
                                                    ;WRITE DATA WITH 60 IN RHCS1

                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA

025370  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
025374  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
025376  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                            ;THE REGISTERS ARE SAVED
025400  000022                        18.                     ;NUMBER OF REGISTERS
                                                            ;SAVED = 18.

025402  004737  041446                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                            ;AND THAT NO STATUS BITS ARE STUCK = 1
025406  104400  066402                TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                            ;THE FIRST SET OF BITS DON'T = 1
025412  000000                        HALT                    ;STOP

025414  013777  004606  154644        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

025422  013746  002442                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
025426  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                            ;ENABLE INTERRUPT
025432  012677  154642                MOV     (SP)+,@RHCS1    ;GO WITH
                                                             ;60 IN RHCS1 FOR WRITE DATA
                                                            ;WITH INTERRUPT ENABLED
025436  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
025440  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

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

```
6041
6042  025442  104412                        WAT                     ;WAIT FOR RDY BIT TO SET
6043  025444  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
6044  025446  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6045  025450  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
6046  025452  001507                        839.                    ;RDY MUST SET BETWEEN
6047                                                                ;690 AND 17470 MICRO SECONDS
6048
6049                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6050                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
6051
6052  025454  013746  002442                MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
6053  025460  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6054  025464  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6055  025470  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6056                                                                ;AND COMMAND SHOULD BE SET
6057  025472  001405                        BEQ     64$             ;BRANCH IF GOOD
6058  025474  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
6059  025500  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6060  025504  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
6061                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
6062  025506  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6063  025512  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6064  025516  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6065                                                                ;SHOULD BE SET
6066  025520  001405                        BEQ     66$             ;BRANCH IF GOOD
6067  025522  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
6068  025526  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6069  025532  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
6070                                                                ;MOL!DPR!VV SHOULD BE SET
6071  025534                        66$:
6072
6073                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6074
6075  025534  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6076  025540  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
6077  025542  000000                        0                       ;DATA
6078  025544  004037  041270                JSR     R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
6079  025550  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
6080  025552  003470                        WRFROM+<256.*2>                 ;DATA
6081  025554  004037  041270                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6082  025560  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
6083  025562  000001                        1                       ;DATA
6084
6085                                        ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6086                                        ;*AFTER COMMAND
6087
6088
6089  025564  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6090                                                                ;PRESENT VALUE
6091  025570  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6092  025572  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
6093  025574  000022                        18.                     ;18. REGISTERS TO BE COMPARED
6094  025576  025602                        1$                      ;RETURN TO 1$ ON ERROR
```

```
6095  025600  025606                          2$                      ;RETURN TO 2$ ON NO ERROR
6096
6097  025602  104035              1$:         ERROR   35              ;WRITE DATA COMMAND CAUSED
6098  025604  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
6099                                                                  ;GOOD DATA GIVES WHAT SHOULD
6100                                                                  ;BE
6101                                                                  ;RECEIVED DATA GIVES WHAT WAS
6102                                                                  ;THERE AFTER COMMAND
6103
6104                                          ;*NOW WRITE FROM BUFFER WILL CHECKED FOR NO CHANGE
6105
6106  025606                      2$:
6107
6108  025606  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
6109  025612  003534                          REINTO                  ;GOOD DATA STARTS FROM REINTO
6110  025614  002470                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
6111  025616  000400                          256.                    ;256. WORDS TO BE COMPARED
6112  025620  025624                          3$                      ;RETURN TO 3$ ON ERROR
6113  025622  025630                          4$                      ;RETURN TO 4$ ON NO ERROR
6114
6115
6116  025624  104036              3$:         ERROR   36              ;WRITE DATA COMMAND CHANGED
6117  025626  000207                          RTS     PC              ;WRITE FROM BUFFER
6118
6119                                          ;*NOW A READ DATA COMMAND WILL BE GIVEN
6120                                          ;*FILL READ INTO BUFFER WITH 256 ZEROS
6121
6122  025630                      4$:
6123
6124  025630  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6125                                                                  ;R3-RHDS1, R4-RHER1
6126                                                                  ;GIVE RH-11 INITIALIZE
6127                                                                  ;SETUP UNIT NUMBER
6128  025634  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
6129  025640  003534                          REINTO                  ;STARTING FROM REINTO
6130  025642  000400                          256.                    ;256. WORDS
6131  025644  000000                          0                       ;FILL WITH 0
6132
6133
6134                                          ;*FILL WRITE FROM BUFFER WITH 256 OF 125252
6135
6136  025646  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
6137  025652  002470                          WRFROM                  ;STARTING FROM WRFROM
6138  025654  000400                          256.                    ;256. WORDS
6139  025656  125252                          125252                  ;FILL WITH 125252
6140
6141
6142                                          ;*NOW FILL COMMAND
6143
6144
6145  025660  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6146  025664  000000                          0                       ;CYLINDER 0
6147  025666     000          .BYTE   0       ;SECTOR 0
6148  025667     000          .BYTE   0       ;TRACK 0
```

```
6149  025670  177400                          -256.                   ;WORD COUNT = 256.
6150  025672  003534                          REINTO                  ;BUS ADDRESS
6151                                                                  ;STARTING ADDRESS OF DATA
6152                                                                  ;BUFFER = REINTO
6153  025674  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6154  025676  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
6155                                                                  ;INHIBIT ECC CORRECTION
6156                                                                  ;DO NOT INHIBIT HEADER COMPARE
6157  025700  002446                          READAT                  ;GET READY TO DO A READAT
6158                                                        ;READ DATA WITH 70 IN RHCS1
6159
6160
6161                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6162
6163  025702  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
6164  025706  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6165  025710  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
6166                                                                  ;THE REGISTERS ARE SAVED
6167  025712  000022                          18.                     ;NUMBER OF REGISTERS
6168                                                                  ;SAVED = 18.
6169
6170  025714  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6171                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
6172  025720  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
6173                                                                  ;THE FIRST SET OF BITS DON'T = 1
6174  025724  000000                          HALT                    ;STOP
6175
6176  025726  013777  004606  154332          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6177                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6178                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6179                                                                  ;'TIME' WILL ONLY SAVE
6180                                                                  ;CURRENT CYLINDER ADDRESS
6181                                                                  ;AND LOOK AHEAD REGISTERS
6182
6183
6184  025734  013746  002446                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
6185  025740  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
6186                                                                  ;ENABLE INTERRUPT
6187  025744  012677  154330                  MOV     (SP)+,@RHCS1    ;GO WITH
6188                                                                   ;70 IN RHCS1 FOR READ DATA
6189                                                                  ;WITH INTERRUPT ENABLED
6190  025750  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
6191  025752  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6192
6193
6194  025754  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
6195  025756  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6196  025760  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6197  025762  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
6198  025764  001507                          839.                    ;RDY MUST SET BETWEEN
6199                                                                  ;690 AND 17470 MICRO SECONDS
6200
6201                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6202                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
```

```
6203
6204  025766  013746  002446                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
6205  025772  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6206  025776  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6207  026002  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6208                                                                  ;AND COMMAND SHOULD BE SET
6209  026004  001405                          BEQ     67$             ;BRANCH IF GOOD
6210  026006  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
6211  026012  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6212  026016  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
6213                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
6214  026020  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6215  026024  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6216  026030  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6217                                                                  ;SHOULD BE SET
6218  026032  001405                          BEQ     69$             ;BRANCH IF GOOD
6219  026034  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
6220  026040  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6221  026044  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
6222                                                                  ;MOL!DPR!VV SHOULD BE SET
6223  026046                          69$:
6224
6225                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
6226
6227  026046  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6228  026052  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
6229  026054  000000                          0                       ;DATA
6230  026056  004037  041270                  JSR     R0,@#FILLRE     ;MOV REINTO+(256.*2) INTO SAVED RHBA
6231  026062  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
6232  026064  004534                          REINTO+(256.*2)                 ;DATA
6233  026066  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6234  026072  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
6235  026074  000001                          1                       ;DATA
6236
6237                                          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
6238                                          ;*WITH REGISTERS AFTER COMMAND
6239
6240
6241  026076  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6242                                                                  ;PRESENT VALUE
6243  026102  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6244  026104  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6245  026106  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6246  026110  026114                          5$                      ;RETURN TO 5$ ON ERROR
6247  026112  026120                          6$                      ;RETURN TO 6$ ON NO ERROR
6248
6249  026114  104033                  5$:     ERROR   33              ;READ DATA CAUSED IMPROPER
6250  026116  000207                          RTS     PC              ;REGISTER CHANGE
6251                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THE
6252                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
6253                                                                  ;AFTER COMMAND
6254
6255                                          ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6256
```

```
026120                          6$:

026120  004037  043434                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
026124  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
026126  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
026130  000400                          256.                    ;256. WORDS TO BE COMPARED
026132  026136                          7$                      ;RETURN TO 7$ ON ERROR
026134  026142                          10$                     ;RETURN TO 10$ ON NO ERROR


026136  104034                  7$:     ERROR   34              ;INCORRECT DATA AFTER
026140  000207                          RTS     PC              ;WRITE DATA FOLLOWED BY A
                                                                ;READ DATA

026142                          10$:
```

```
                        ;;*****************************************************************
                        ;*TEST 37       WRITE/READ DATA (052525)
                        ;*
                        ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                        ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                        ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                        ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                        ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                        ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                        ;*      200 OF 052525 AND 56 OF 377
                        ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                        ;*      ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATI
                        ;*
                        ;*
                        ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                        ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                        ;*
                        ;*      THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                        ;*      AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                        ;*      WORDS OF 52525 AND 56 WORDS OF 0
                        ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                        ;*      GO IS GIVEN
                        ;*
                        ;*
                        ;*      ALL REGISTER ARE CHECKED
                        ;*      READ DATA IS CHECKED
                        ;;*****************************************************************
026142  000004          ↑ST37:  SCOPE
026144  012706  001000          MOV     #STACK,SP       ;RESET STACK
026150  012737  000037  004604  MOV     #37,@#↑STNM     ;SAVE TEST NUMBER
026156  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
                        ;*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
026162  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
026166  002470                  WRFROM                  ;STARTING FROM WRFROM
026170  000310                  200.                    ;200. WORDS
026172  052525                  52525                   ;FILL WITH 52525
026174  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
026200  003310                  WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
026202  000070                  56.                     ;56. WORDS
026204  000377                  377                     ;FILL WITH 377
                        ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
                        ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                        ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
```

```
 6326  026206  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
 6327  026212  003534                          REINTO                  ;STARTING FROM REINTO
 6328  026214  000310                          200.                    ;200. WORDS
 6329  026216  052525                          52525                   ;FILL WITH 52525
 6330
 6331  026220  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
 6332  026224  004354                          REINTO+(200.*2)                 ;STARTING FROM REINTO+(200.*2)
 6333  026226  000070                          56.                     ;56. WORDS
 6334  026230  000377                          377                     ;FILL WITH 377
 6335
 6336
 6337                                          ;*NOW WRITE DATA COMMAND WILL BE LOADED
 6338
 6339
 6340  026232  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 6341  026236  000000                          0                       ;CYLINDER 0
 6342  026240     000                  .BYTE   0                       ;SECTOR 0
 6343  026241     000                  .BYTE   0                       ;TRACK 0
 6344  026242  177470                          -200.                   ;WORD COUNT = 200.
 6345  026244  002470                          WRFROM                  ;BUS ADDRESS
 6346                                                                  ;STARTING ADDRESS OF DATA
 6347                                                                  ;BUFFER = WRFROM
 6348  026246  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 6349  026250  010000                          FMT22                   ;16 BITS PER WORD FORMAT
 6350                                                                  ;DO NOT INHIBIT ECC CORRECTION
 6351                                                                  ;DO NOT INHIBIT HEADER COMPARE
 6352  026252  002442                          WRIDAT                  ;GET READY TO DO A WRIDAT
 6353                                                  ;WRITE DATA WITH 60 IN RHCS1
 6354
 6355
 6356                                          ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
 6357
 6358  026254  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
 6359  026260  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 6360  026262  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
 6361                                                                  ;THE REGISTERS ARE SAVED
 6362  026264  000022                          18.                     ;NUMBER OF REGISTERS
 6363                                                                  ;SAVED = 18.
 6364
 6365  026266  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 6366                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
 6367  026272  104400  066432                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 6368                                                                  ;THE FIRST SET OF BITS DON'T = 1
 6369  026276  000000                          HALT                    ;STOP
 6370
 6371  026300  013777  004606  153760          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 6372                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 6373                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 6374                                                                  ;'TIME' WILL ONLY SAVE
 6375                                                                  ;CURRENT CYLINDER ADDRESS
 6376                                                                  ;AND LOOK AHEAD REGISTERS
 6377
 6378
 6379  026306  013746  002442                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
```

```
6390  026312  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
6391                                                          ;ENABLE INTERRUPT
6392  026316  012677  153756          MOV     (SP)+,@RHCS1    ;GO WITH
6393                                                           ;GO IN RHCS1 FOR WRITE DATA
6394                                                          ;WITH INTERRUPT ENABLED
6395  026322  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
6396  026324  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6397
6398                                  ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6399
6400
6401  026326  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
6402  026330  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
6403  026332  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6404  026334  001614                  900.                    ;ALLOW 9080 MICRO SECONDS
6405  026336  001507                  939.                    ;RDY MUST SET BETWEEN
6406                                                          ;690 AND 17470 MICRO SECONDS
6407
6408                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6409                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
6410
6411  026340  013746  002442          MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
6412  026344  052716  004101          BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6413  026350  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6414  026354  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6415                                                          ;AND COMMAND SHOULD BE SET
6416  026356  001405                  BEQ     64$             ;BRANCH IF GOOD
6417  026360  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
6418  026364  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6419  026370  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
6420                                                          ;COMMAND AND IE!GO!DVA SHOULD BE SET
6421  026372  012746  010500  64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6422  026376  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6423  026402  022605                  CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6424                                                          ;SHOULD BE SET
6425  026404  001405                  BEQ     66$             ;BRANCH IF GOOD
6426  026406  010537  001126          MOV     R5,@#$BDDAT     ;BAD DATA
6427  026412  010337  004600          MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6428  026416  104063                  ERROR   63              ;DURING ABOVE OPERATION ONLY
6429                                                          ;MOL!DPR!VV SHOULD BE SET
6430  026420                  66$:
6431
6432                                  ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6433
6434  026420  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6435  026424  002272                  RHWC                    ;SAVED REGISTER TO CHANGE
6436  026426  000000                  0                       ;DATA
6437  026430  004037  041270          JSR     R0,@#FILLRE     ;MOV WRFROM+(200.*2) INTO SAVED RHBA
6438  026434  002274                  RHBA                    ;SAVED REGISTER TO CHANGE
6439  026436  003310                  WRFROM+(200.*2)                 ;DATA
6440  026440  004037  041270          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6441  026444  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
6442  026446  000001                  1                       ;DATA
6443
```

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

026450  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
026454  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
026456  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
026460  000022                          18.                     ;18. REGISTERS TO BE COMPARED
026462  026466                          1$                      ;RETURN TO 1$ ON ERROR
026464  026472                          2$                      ;RETURN TO 2$ ON NO ERROR

026466  104035                  1$:     ERROR   35              ;WRITE DATA COMMAND CAUSED
026470  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER COMMAND

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

026472                          2$:

026472  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
026476  003534                          REINTO                  ;GOOD DATA STARTS FROM REINTO
026500  002470                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
026502  000400                          256.                    ;256. WORDS TO BE COMPARED
026504  026510                          3$                      ;RETURN TO 3$ ON ERROR
026506  026514                          4$                      ;RETURN TO 4$ ON NO ERROR

026510  104036                  3$:     ERROR   36              ;WRITE DATA COMMAND CHANGED
026512  000207                          RTS     PC              ;WRITE FROM BUFFER

                                        ;*NOW A READ DATA COMMAND WILL BE GIVEN

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

026514                          4$:

026514  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
026520  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
026524  003534                          REINTO                  ;STARTING FROM REINTO
026526  000310                          200.                    ;200. WORDS
026530  000000                          0                       ;FILL WITH 0

026532  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
026536  004354                          REINTO+(200.*2)                 ;STARTING FROM REINTO+(200.*2)
026540  000070                          56.                     ;56. WORDS
026542  000377                          377                     ;FILL WITH 377
```

```
                                        ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
                                        JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
026550  002470                          WRFROM                  ;STARTING FROM WRFROM
026552  000310                          200.                    ;200. WORDS
026554  052525                          52525                   ;FILL WITH 52525
                                        JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
026562  003310                          WRFROM+<200.*2>         ;STARTING FROM WRFROM+<200.*2>
026564  000070                          56.                     ;56. WORDS
026566  000377                          377                     ;FILL WITH 377

                                        ;*NOW FILL COMMAND

026570  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
026574  000000                          0                       ;CYLINDER 0
026576     000                  .BYTE   0                       ;SECTOR 0
026577     000                  .BYTE   0                       ;TRACK 0
026600  177470                          -200.                   ;WORD COUNT = 200.
026602  003534                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
026604  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
026606  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
026610  002446                          READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

026612  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
026616  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
026620  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
026622  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

026624  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
026630  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
026634  000000                          HALT                    ;STOP

026636  013777  004606  153422          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
```

```
                                                ;GET READY TO MOVE COMMAND
026644  013746  002446          MOV     @#READAT,-(SP)
026650  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                ;ENABLE INTERRUPT
026654  012677  153420          MOV     (SP)+,@RHCS1    ;GO WITH
                                                ; 70 IN RHCS1 FOR READ DATA
                                                ;WITH INTERRUPT ENABLED
026660  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
026662  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

026664  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
026666  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
026670  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
026672  001614                  908.                    ;ALLOW 9080 MICRO SECONDS
026674  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

026676  013746  002446          MOV     @#READAT,-(SP)  ;SAVE COMMAND
026702  052716  004101          BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
026706  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
026712  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                ;AND COMMAND SHOULD BE SET
026714  001405                  BEQ     67$             ;BRANCH IF GOOD
026716  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
026722  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
026726  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
026730  012746  010500  67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
026734  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
026740  022605                  CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                ;SHOULD BE SET
026742  001405                  BEQ     69$             ;BRANCH IF GOOD
026744  010537  001126          MOV     R5,@#$BDDAT     ;BAD DATA
026750  010337  004600          MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
026754  104063                  ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                ;MOL!DPR!VV SHOULD BE SET
026756                  69$:

                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

026756  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
026762  002272                  RHWC                    ;SAVED REGISTER TO CHANGE
026764  000000                  0                       ;DATA
026766  004037  041270          JSR     R0,@#FILLRE     ;MOV REINTO+<200.*2> INTO SAVED RHBA
026772  002274                  RHBA                    ;SAVED REGISTER TO CHANGE
026774  004354                  REINTO+<200.*2>                 ;DATA
026776  004037  041270          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
027002  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
027004  000001                  1                       ;DATA
```

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

027006  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                              ;PRESENT VALUE
027012  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
027014  002354                        W0                      ;TEST DATA STARTING FROM 'RHW0'
027016  000022                        19.                     ;18. REGISTERS TO BE COMPARED
027020  027024                        5$                      ;RETURN TO 5$ ON ERROR
027022  027030                        6$                      ;RETURN TO 6$ ON NO ERROR

027024  104033            5$:         ERROR   33              ;READ DATA CAUSED IMPROPER
027026  000207                        RTS     PC              ;REGISTER CHANGE
                                                              ;GOOD DATA GIVES WHAT SHOULD BE THE
                                                              ;RECEIVED DATA GIVES WHAT WAS THERE
                                                              ;AFTER COMMAND

                                        ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

027030            6$:

027030  004037  043404                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
027034  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
027036  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
027040  000400                        256.                    ;256. WORDS TO BE COMPARED
027042  027046                        7$                      ;RETURN TO 7$ ON ERROR
027044  027052                        10$                     ;RETURN TO 10$ ON NO ERROR

027046  104034            7$:         ERROR   34              ;INCORRECT DATA AFTER
027050  000207                        RTS     PC              ;WRITE DATA FOLLOWED BY A
                                                              ;READ DATA
027052            10$:
```

```
                                ;;*****************************************************************
                                ;*TEST 40       WRITE/READ DATA USING UNIBUS B
                                ;
                                ;*      THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
                                ;*      IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
                                ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                                ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                                ;*      THESE COMMANDS USE UNIBUS B FOR DATA
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      200 OF 052525 AND 56 OF 377
                                ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
                                ;*
                                ;*
                                ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                                ;*
                                ;*      THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                                ;*      AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                                ;*      WORDS OF 52525 AND 56 WORDS OF 0
                                ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN
                                ;*
                                ;*
                                ;*      ALL REGISTER ARE CHECKED
                                ;*      READ DATA IS CHECKED

                                ;;*****************************************************************
027052  000004                  TST40:  SCOPE
027054  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
027062  012706  001000                  MOV     #STACK,SP       ;RESET STACK
027066  012737  000040  004604          MOV     #40,@#TSTNM     ;SAVE TEST NUMBER

027074  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

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

027100  005737  004750                  TST     @#RH70          ;TEST FOR RH70 CONTROLLER
027104  001402                  BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
027106  000137  030222          JMP     TST41   ;       ;JUMP TO NEXT TEST ------------------
027112                          30$:                    ;IF FLAG = 1, DO THIS TEST

027112  005037  004732                  CLR     @#UBUSB         ;CLEAR UNIBUS INDICATOR

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

027116  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
027122  002470                  WRFROM                  ;STARTING FROM WRFROM
```

```
6689  027124  000310                          200.                    ;200. WORDS
6690  027126  052525                          52525                   ;FILL WITH 52525
6691
6692  027130  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6693  027134  003310                          WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
6694  027136  000070                          56.                     ;56. WORDS
6695  027140  000377                          377                     ;FILL WITH 377
6696
6697
6698                                          ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
6699                                          ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
6700                                          ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
6701
6702  027142  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
6703  027146  003534                          REINTO                  ;STARTING FROM REINTO
6704  027150  000310                          200.                    ;200. WORDS
6705  027152  052525                          52525                   ;FILL WITH 52525
6706
6707  027154  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6708  027160  004354                          REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
6709  027162  000070                          56.                     ;56. WORDS
6710  027164  000377                          377                     ;FILL WITH 377
6711
6712
6713                                          ;*NOW WRITE DATA COMMAND WILL BE LOADED
6714
6715
6716  027166  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6717  027172  000000                          0                       ;CYLINDER 0
6718  027174     000                  .BYTE   0                       ;SECTOR 0
6719  027175     000                  .BYTE   0                       ;TRACK 0
6720  027176  177470                          -200.                   ;WORD COUNT = 200.
6721  027200  002470                          WRFROM                  ;BUS ADDRESS
6722                                                                  ;STARTING ADDRESS OF DATA
6723                                                                  ;BUFFER = WRFROM
6724  027202  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6725  027204  010000                          FMT22                   ;16 BITS PER WORD FORMAT
6726                                                                  ;DO NOT INHIBIT ECC CORRECTION
6727                                                                  ;DO NOT INHIBIT HEADER COMPARE
6728  027206  002442                          WRIDAT                  ;GET READY TO DO A WRIDAT
6729                                                  ;WRITE DATA WITH 60 IN RHCS1
6730
6731  027210  052777  002000  153062          BIS     #PSEL,@RHCS1    ;SET PORT B
6732                                                                  ;THAT IS UNIBUS B
6733
6734                                          ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
6735
6736  027216  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
6737  027222  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6738  027224  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
6739                                                                  ;THE REGISTERS ARE SAVED
6740  027226  000022                          18.                     ;NUMBER OF REGISTERS
6741                                                                  ;SAVED = 18.
6742
```

```
6743  027230  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6744                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
6745  027234  104400  056402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
6746                                                                  ;THE FIRST SET OF BITS DON'T = 1
6747  027240  000000                          HALT                    ;STOP
6748
6749  027242  013777  004606  153016          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6750                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6751                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6752                                                                  ;'TIME' WILL ONLY SAVE
6753                                                                  ;CURRENT CYLINDER ADDRESS
6754                                                                  ;AND LOOK AHEAD REGISTERS
6755
6756  027250  013746  002442                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
6757  027254  052716  002101                  BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
6758                                                                  ;ENABLE INTERRUPT
6759  027260  012677  153014                  MOV     (SP)+,@RHCS1    ;GO WITH
6760                                                                  ;GO IN RHCS1 FOR WRITE DATA
6761                                                                  ;WITH INTERRUPT ENABLED
6762  027264  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
6763  027266  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6764
6765                                          ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6766
6767
6768  027270  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
6769  027272  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6770  027274  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6771  027276  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
6772  027300  001507                          839.                    ;RDY MUST SET BETWEEN
6773                                                                  ;690 AND 17470 MICRO SECONDS
6774
6775                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6776                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
6777
6778  027302  013746  002442                  MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
6779  027306  052716  006101                  BIS     #IE!GO!DVA!PSEL,(SP)    ;INCLUDE IE!GO!DVA!PSEL
6780  027312  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6781  027316  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
6782                                                                  ;AND COMMAND SHOULD BE SET
6783  027320  001405                          BEQ     64$             ;BRANCH IF GOOD
6784  027322  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
6785  027326  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6786  027332  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
6787                                                                  ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
6788  027334  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6789  027340  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6790  027344  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6791                                                                  ;SHOULD BE SET
6792  027346  001405                          BEQ     66$             ;BRANCH IF GOOD
6793  027350  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
6794  027354  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6795  027360  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
6796                                                                  ;MOL!DPR!VV SHOULD BE SET
```

```
  6797  027362                                  66$:
  6798
  6799
  6800                                          ;*CHECK IF NEM NON EXISTANT MEMORY IS SET
  6801                                          ;*IF SET IT MEANS UNIBUS B IS NOT CONNECTED
  6802                                          ;*SO THIS TEST IS NOT PERFORMED
  6803
  6804  027362  032777  004000  152706          BIT     #NEM,@RHCS2     ;TEST NEM
  6805  027370  001441                          BEQ     11$             ;BRANCH IF UNIBUS B THERE
  6806  027372  012737  177777  004732          MOV     #-1,@#UBUSB     ;UNIBUS B NOT THERE
  6807  027400  104400  027406                  TYPE    ,68$            ;;TYPE ASCIZ STRING
  6808  027404  000425                          BR      67$             ;;GET OVER THE ASCIZ
  6809                                  ;;68$:  .ASCIZ  <15><12>/THE RH DOES NOT HAVE UNIBUS B CONNECTED
  6810  027460                          67$:
  6811  027460  104400  001223                  TYPE    ,$CRLF
  6812  027464  104400  001223                  TYPE    ,$CRLF
  6813  027470  000137  030222                  JMP     @#10$           ;JUMP TO NEXT TEST - NO UNIBUS B
  6814  027474                          11$:
  6815  027474  104400  027502                  TYPE    ,70$            ;;TYPE ASCIZ STRING
  6816  027500  000424                          BR      69$             ;;GET OVER THE ASCIZ
  6817                                  ;;70$:  .ASCIZ  <15><12>/THE RH DOES HAVE UNI BUS B CONNECTED
  6818  027552                          69$:
  6819  027552  104400  001223                  TYPE    ,$CRLF
  6820  027556  104400  001223                  TYPE    ,$CRLF
  6821
  6822                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
  6823
  6824  027562  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
  6825  027566  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
  6826  027570  000000                          0                       ;DATA
  6827  027572  004037  041270                  JSR     R0,@#FILLRE     ;MOV WRFROM+<200.*2> INTO SAVED RHBA
  6828  027576  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
  6829  027600  003310                          WRFROM+<200.*2>                 ;DATA
  6830  027602  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
  6831  027606  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
  6832  027610  000001                          1                       ;DATA
  6833
  6834                                          ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
  6835                                          ;*AFTER COMMAND
  6836
  6837
  6838  027612  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
  6839                                          ;PRESENT VALUE
  6840  027616  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
  6841  027620  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
  6842  027622  000022                          18.                     ;18. REGISTERS TO BE COMPARED
  6843  027624  027630                          1$                      ;RETURN TO 1$ ON ERROR
  6844  027626  027634                          2$                      ;RETURN TO 2$ ON NO ERROR
  6845
  6846  027630  104074                  1$:     ERROR   74              ;WHILE USING UNIBUS B
  6847                                          ;WRITE DATA COMMAND CAUSED
  6848  027632  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
  6849                                          ;GOOD DATA GIVES WHAT SHOULD
  6850                                          ;BE
```

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

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

027634                                  3$:

027634  004037  041404                        JSR     R0,$$COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
027640  027700                        REINTO          ;GOOD DATA STARTS FROM REINTO
027642  027756                        WRFROM          ;"TES" DATA STARTS FROM WRFROM
027644  000256                        256.            ;256. WORDS TO BE COMPARED
027646  027660                        3$              ;RETURN TO 3$ ON ERROR
027650  027662                        4$              ;RETURN TO 4$ ON NO ERROR

027652  104075                  3$:   ERROR   75      ;WHILE USING UNIBUS B
                                                        ;WRITE DATA COMMAND CHANGED
027654  000207                        RTS     PC      ;WRITE FROM BUFFER

                                                ;*NOW A READ DATA COMMAND WILL BE GIVEN
                                                ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES

027656                                  4$:

027656  004737  041366                        JSR     PC,$$CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
027662  004037  041236                        JSR     R0,$$CLAREA     ;CLEAR 200. WORDS, FROM REINTO
027666  003534                        REINTO          ;STARTING FROM REINTO
027670  000310                        200.            ;200. WORDS
027672  000000                        0               ;FILL WITH 0

027674  004037  041236                        JSR     R0,$$CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
027700  004354                        REINTO+(200.*2)                 ;STARTING FROM REINTO+(200.*2)
027702  000070                        56.             ;56. WORDS
027704  000377                        377             ;FILL WITH 377

                                                ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0

027706  004037  041236                        JSR     R0,$$CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
027712  002470                        WRFROM          ;STARTING FROM WRFROM
027714  000310                        200.            ;200. WORDS
027716  052525                        52525           ;FILL WITH 52525

027720  004037  041236                        JSR     R0,$$CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+(200.*2)
027724  003310                        WRFROM+(200.*2)                 ;STARTING FROM WRFROM+(200.*2)
027726  000070                        56.             ;56. WORDS
027730  000377                        377             ;FILL WITH 377

                                                ;*NOW FILL COMMAND
```

```
                                                ;SETUP TO RUN FOR DATA COMMAND
                                                ;CYLINDER 0
                                .BYTE   0       ;SECTOR 0
                                .BYTE   0       ;TRACK 0
                                -200.           ;WORD COUNT = 200.
                                REINTO          ;BUS ADDRESS
                                                ;STARTING ADDRESS OF DATA
                                                ;BUFFER = REINTO
                                0               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
                                ECC!FMT22               ;16 BITS PER WORD FORMAT
                                                ;INHIBIT ECC CORRECTION
                                                ;DO NOT INHIBIT HEADER COMPARE
                                READAT          ;GET READY TO DO A READAT
                                        ;READ DATA WITH 70 IN RHCS1

                                BIS     #PSEL,@RHCS1    ;SET PORT B
                                                ;THAT IS UNIBUS B

                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

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

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

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

                                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
                                BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
                                                ;ENABLE INTERRUPT
                                MOV     (SP)+,@RHCS1    ;GO WITH
                                                ; ;70 IN RHCS1 FOR READ DATA
                                                ;WITH INTERRUPT ENABLED
                                MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
                                MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

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

```
006959  030042  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
006960  030044  001507                          839.                    ;RDY MUST SET BETWEEN
006961                                                                  ;690 AND 17470 MICRO SECONDS
006962
006963                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
006964                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
006965
006966  030046  013746  002446                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
006967  030052  052716  006101                  BIS     #IE!GO!DVA!PSEL,(SP)    ;INCLUDE IE!GO!DVA!PSEL
006968  030056  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
006969  030062  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
006970                                                                  ;AND COMMAND SHOULD BE SET
006971  030064  001405                          BEQ     71$             ;BRANCH IF GOOD
006972  030066  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
006973  030072  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
006974  030076  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
006975                                                                  ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
006976  030100  012746  010500          71$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
006977  030104  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
006978  030110  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
006979                                                                  ;SHOULD BE SET
006980  030112  001405                          BEQ     73$             ;BRANCH IF GOOD
006981  030114  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
006982  030120  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
006983  030124  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
006984                                                                  ;MOL!DPR!VV SHOULD BE SET
006985  030126                          73$:
006986
006987                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
006988
006989  030126  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
006990  030132  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
006991  030134  000000                          0                       ;DATA
006992  030136  004037  041270                  JSR     R0,@#FILLRE     ;MOV REINTO+(200.*2) INTO SAVED RHBA
006993  030142  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
006994  030144  004354                          REINTO+(200.*2)                 ;DATA
006995  030146  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
006996  030152  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
006997  030154  000001                          1                       ;DATA
006998
006999                                          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
007000                                          ;*WITH REGISTERS AFTER COMMAND
007001
007002
007003  030156  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
007004                                                                  ;PRESENT VALUE
007005  030162  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
007006  030164  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
007007  030166  000022                          18.                     ;18. REGISTERS TO BE COMPARED
007008  030170  030174                          5$                      ;RETURN TO 5$ ON ERROR
007009  030172  030200                          6$                      ;RETURN TO 6$ ON NO ERROR
007010
007011  030174  104072                  5$:     ERROR   72              ;WHILE USING UNIBUS B
007012                                                                  ;READ DATA CAUSED IMPROPER
```

```
 7013  030176  000207                          RTS     PC              ;REGISTER CHANGE
 7014                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THE
 7015                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
 7016                                                                  ;AFTER COMMAND
 7017
 7018                                          ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
 7019
 7020  030200                          6$:
 7021
 7022  030200  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 7023  030204  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 7024  030206  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
 7025  030210  000400                          256.                    ;256. WORDS TO BE COMPARED
 7026  030212  030216                          7$                      ;RETURN TO 7$ ON ERROR
 7027  030214  030222                          10$                     ;RETURN TO 10$ ON NO ERROR
 7028
 7029
 7030  030216  104073                  7$:     ERROR   73              ;WHILE USING UNIBUS B
 7031                                                                  ;INCORRECT DATA AFTER
 7032  030220  000207                          RTS     PC              ;WRITE DATA FOLLOWED BY A
 7033                                                                  ;READ DATA
 7034  030222                          10$:
 7035
 7036
 7037
 7038
 7039
```

```
                        ;;*****************************************************************
                        ;*TEST 41        IMPLIED SEARCH
                        ;*
                        ;*      ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEARCH
                        ;*      A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTOR
                        ;*      OF WORDS
                        ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                        ;*      TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS
                        ;*      266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1
                        ;*      THEN READ HEADER AND DATA FOR ABOVE SECTOR 1 ONLY
                        ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                        ;*      10000,0,0,0 AND 256 OF 0, 10000,1,0,0, AND 2 OF 1
                        ;*      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 CHANGE
                        ;*      THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
                        ;*
                        ;*
                        ;*      NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
                        ;*      WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000,1,0,0,1,1
                        ;*      AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT
                        ;*      GO AND IE THEN ALL REGISTERS ARE SAVED
                        ;*      GO IS GIVEN FOR THE READ COMMAND, 256 WORDS
                        ;*
                        ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                        ;*      THEN THE READ DATA IS COMPARED
                        ;*
                        ;;*****************************************************************
030222  000004         TST41:  SCOPE
030224  012706  001000          MOV     #STACK,SP       ;RESET STACK
030230  012737  000041  004604  MOV     #41,@#TSTNM     ;SAVE TEST NUMBER
                        ;
030236  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
                        ;
                                ;*FILL WRITE FROM BUFFER WITH HEADER
                        ;
030242  004037  041212          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
030246  002470                  WRFROM                  ;LOCATION WHERE SAVED
030250  000004                  4                       ;NUMBER OF WORDS SAVED
030252  010000                  10000                   ;FIRST DATA WORD
030254  000000                  0                       ;SECOND DATA WORD
030256  000000                  0                       ;THIRD DATA WORD
030260  000000                  0                       ;FOURTH DATA WORD
                        ;
                                ;*FILL WRITE FROM BUFFER WITH DATA
030262  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
030266  002500                  WRFROM+10               ;STARTING FROM WRFROM+10
030270  000400                  256.                    ;256. WORDS
```

```
7095  030272  000000                    0                       ;FILL WITH 0
7096
7097
7098                                    ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
7099
7100  030274  004037  041212            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
7101  030300  003500                    WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
7102  030302  000004                    4                       ;NUMBER OF WORDS SAVED
7103  030304  010000                    10000                   ;FIRST DATA WORD
7104  030306  000001                    1                       ;SECOND DATA WORD
7105  030310  000000                    0                       ;THIRD DATA WORD
7106  030312  000000                    0                       ;FOURTH DATA WORD
7107
7108                                    ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
7109  030314  004037  041236            JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
7110  030320  003510                    WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
7111  030322  000002                    2                       ;2 WORDS
7112  030324  000001                    1                       ;FILL WITH 1
7113
7114
7115                                    ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
7116                                    ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
7117                                    ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
7118                                    ;*CHANGE WRITE FROM BUFFER.
7119
7120
7121  030326  004037  041212            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
7122  030332  003534                    REINTO                  ;LOCATION WHERE SAVED
7123  030334  000004                    4                       ;NUMBER OF WORDS SAVED
7124  030336  010000                    10000                   ;FIRST DATA WORD
7125  030340  000000                    0                       ;SECOND DATA WORD
7126  030342  000000                    0                       ;THIRD DATA WORD
7127  030344  000000                    0                       ;FOURTH DATA WORD
7128  030346  004037  041236            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
7129  030352  003544                    REINTO+10               ;STARTING FROM REINTO+10
7130  030354  000400                    256.                    ;256. WORDS
7131  030356  000000                    0                       ;FILL WITH 0
7132
7133
7134  030360  004037  041212            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO+<260.*2>
7135  030364  004544                    REINTO+<260.*2>                 ;LOCATION WHERE SAVED
7136  030366  000004                    4                       ;NUMBER OF WORDS SAVED
7137  030370  010000                    10000                   ;FIRST DATA WORD
7138  030372  000001                    1                       ;SECOND DATA WORD
7139  030374  000000                    0                       ;THIRD DATA WORD
7140  030376  000000                    0                       ;FOURTH DATA WORD
7141  030400  004037  041236            JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
7142  030404  004554                    REINTO+<264.*2>                 ;STARTING FROM REINTO+<264.*2>
7143  030406  000002                    2                       ;2 WORDS
7144  030410  000001                    1                       ;FILL WITH 1
7145
7146
7147                                    ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7148
```

```
7149
7150  030412  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7151  030416  000000                          0                       ;CYLINDER 0
7152  030420     000                .BYTE     0                       ;SECTOR 0
7153  030421     000                .BYTE     0                       ;TRACK 0
7154  030422  177366                          -262.-4                 ;WORD COUNT (DATA) = 262. +
7155                                                                  ;4 HEADER WORDS
7156  030424  002470                          WRFROM                  ;BUS ADDRESS
7157                                                                  ;STARTING ADDRESS OF DATA
7158                                                                  ;BUFFER = WRFROM
7159  030426  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7160  030430  010000                          FMT22                   ;16 BITS PER WORD FORMAT
7161                                                                  ;DO NOT INHIBIT ECC CORRECTION
7162                                                                  ;DO NOT INHIBIT HEADER COMPARE
7163  030432  002444                          WRIFOR                  ;GET READY TO DO A WRIFOR
7164                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7165
7166
7167                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DAT1
7168  030434  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
7169  030440  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7170  030442  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
7171                                                                  ;THE REGISTERS ARE SAVED
7172  030444  000022                          18.                     ;NUMBER OF REGISTERS
7173                                                                  ;SAVED = 18.
7174
7175  030446  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7176                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
7177  030452  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7178                                                                  ;THE FIRST SET OF BITS DON'T = 1
7179  030456  000000                          HALT                    ;STOP
7180
7181  030460  013777  004606  151600          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7182                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
7183                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7184                                                                  ;'TIME' WILL ONLY SAVE
7185                                                                  ;CURRENT CYLINDER ADDRESS
7186                                                                  ;AND LOOK AHEAD REGISTERS
7187
7188
7189  030466  013746  002444                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
7190  030472  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7191                                                                  ;ENABLE INTERRUPT
7192  030476  012677  151576                  MOV     (SP)+,@RHCS1    ;GO WITH
7193                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7194                                                                  ;WITH INTERRUPT ENABLED
7195  030502  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7196  030504  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7197
7198                                          ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
7199
7200
7201  030506  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
7202  030510  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
7203  030512  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7204  030514  002001                        1025.                   ;ALLOW 10250 MICRO SECONDS
7205  030516  001553                        8.S.                    ;RDY MUST SET BETWEEN
7206                                                                  ;1500 AND 19000 MICRO SECONDS
7207
7208                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7209                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
7210
7211  030520  013746  002444                MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
7212  030524  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7213  030530  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7214  030534  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7215                                                                ;AND COMMAND SHOULD BE SET
7216  030536  001405                        BEQ     64$             ;BRANCH IF GOOD
7217  030540  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
7218  030544  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
7219  030550  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
7220                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
7221  030552  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
7222  030556  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7223  030562  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7224                                                                ;SHOULD BE SET
7225  030564  001405                        BEQ     66$             ;BRANCH IF GOOD
7226  030566  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
7227  030572  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
7228  030576  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
7229                                                                ;MOL!DPR!VV SHOULD BE SET
7230  030600                        66$:
7231
7232                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7233
7234  030600  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
7235  030604  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
7236  030606  000000                        0                       ;DATA
7237  030610  004037  041270                JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
7238  030614  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
7239  030616  003514                        WRFROM+<266.*2>                 ;DATA
7240  030620  004037  041270                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
7241  030624  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
7242  030626  000002                        2                       ;DATA
7243
7244                                        ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
7245                                        ;*WITH REGISTERS AFTER COMMAND
7246
7247
7248  030630  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7249                                                                ;PRESENT VALUE
7250  030634  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7251  030636  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
7252  030640  000022                        18.                     ;18. REGISTERS TO BE COMPARED
7253  030642  030646                        1$                      ;RETURN TO 1$ ON ERROR
7254  030644  030652                        2$                      ;RETURN TO 2$ ON NO ERROR
7255
7256  030646  104027                1$:     ERROR   27              ;WRITE HEADER AND DATA
```

```
7257  030650  000207                          RTS     PC              ;CAUSED IMPROPER REGISTER
7258                                                                  ;CHANGE
7259                                                                  ;GOOD DATA GIVES WHAT SHOULD
7260                                                                  ;BE THERE
7261                                                                  ;RECEIVED DATA GIVES WHAT
7262                                                                  ;WAS THERE AFTER COMMANT
7263
7264                                          ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
7265                                          ;*NOTHING GOT CHANGED
7266
7267  030652                          2$:
7268
7269  030652  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7270  030656  003534                          REINTO                  ;GOOD DATA STARTS FROM REINTO
7271  030660  002470                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
7272  030662  000412                          266.                    ;266. WORDS TO BE COMPARED
7273  030664  030670                          3$                      ;RETURN TO 3$ ON ERROR
7274  030666  030674                          4$                      ;RETURN TO 4$ ON NO ERROR
7275
7276
7277  030670  104030                  3$:     ERROR   30              ;WRITE HEADER AND DATA
7278  030672  000207                          RTS     PC              ;CHANGED WRITE FROM BUFFER
7279
7280                                          ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
7281                                          ;*FOR SECTOR 1, 256 WORDS
7282                                          ;*READ INTO BUFFER IS FILLED WITH ONES
7283  030674                          4$:
7284
7285  030674  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7286                                                                  ;R3-RHDS1, R4-RHER1
7287                                                                  ;GIVE RH-11 INITIALIZE
7288                                                                  ;SETUP UNIT NUMBER
7289  030700  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
7290  030704  003534                          REINTO                  ;STARTING FROM REINTO
7291  030706  000404                          260.                    ;260. WORDS
7292  030710  000377                          377                     ;FILL WITH 377
7293
7294
7295                                          ;*WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA
7296
7297
7298  030712  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
7299  030716  002470                          WRFROM                  ;LOCATION WHERE SAVED
7300  030720  000006                          6                       ;NUMBER OF WORDS SAVED
7301  030722  010000                          10000                   ;FIRST DATA WORD
7302  030724  000001                          1                       ;SECOND DATA WORD
7303  030726  000000                          0                       ;THIRD DATA WORD
7304  030730  000000                          0                       ;FOURTH DATA WORD
7305  030732  000001                          1                       ;FIFTH DATA WORD
7306  030734  000001                          1                       ;SIXTH DATA WORD
7307  030736  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 254 WORDS, FROM WRFROM+<6.*2>
7308  030742  002504                          WRFROM+<6.*2>                   ;STARTING FROM WRFROM+<6.*2>
7309  030744  000254                          254                     ;254 WORDS
7310  030746  000000                          0                       ;FILL WITH 0
```

```
7311
7312
7313                                        ;*NOW FILL COMMAND
7314
7315
7316  030750  004037  043340                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7317  030754  000000                        0                       ;CYLINDER 0
7318  030756     001              .BYTE     1                       ;SECTOR 1
7319  030757     000              .BYTE     0                       ;TRACK 0
7320  030760  177374                        -256.-4                 ;WORD COUNT (DATA) = 256. +
7321                                                                ;4 HEADER WORDS
7322  030762  003534                        REINTO                  ;BUS ADDRESS
7323                                                                ;STARTING ADDRESS OF DATA
7324                                                                ;BUFFER = REINTO
7325  030764  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7326  030766  014000                        ECI!FMT22                       ;16 BITS PER WORD FORMAT
7327                                                                ;INHIBIT ECC CORRECTION
7328                                                                ;DO NOT INHIBIT HEADER COMPARE
7329  030770  002450                        REFOR                   ;GET READY TO DO A REFOR
7330                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1
7331
7332
7333                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7334
7335  030772  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
7336  030776  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7337  031000  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
7338                                                                ;THE REGISTERS ARE SAVED
7339  031002  000022                        18.                     ;NUMBER OF REGISTERS
7340                                                                ;SAVED = 18.
7341
7342  031004  004737  041446                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7343                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
7344  031010  104400  066402                TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7345                                                                ;THE FIRST SET OF BITS DON'T = 1
7346  031014  000000                        HALT                    ;STOP
7347
7348  031016  013777  004606  151242        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7349                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
7350                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7351                                                                ;'TIME' WILL ONLY SAVE
7352                                                                ;CURRENT CYLINDER ADDRESS
7353                                                                ;AND LOOK AHEAD REGISTERS
7354
7355
7356  031024  013746  002450                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
7357  031030  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7358                                                                ;ENABLE INTERRUPT
7359  031034  012677  151240                MOV     (SP)+,@RHCS1    ;GO WITH
7360                                                                  ;72 IN RHCS1 FOR READ DATA
7361                                                                ;WITH INTERRUPT ENABLED
7362  031040  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7363  031042  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7364
```

```
7365                                        ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR 760 MICRO SECONDS
7366
7367
7368  031044  104412                        WAT                 ;WAIT FOR RDY BIT TO SET
7369  031046  002300                        RHCS1               ;WAIT FOR RHCS1 REGISTER
7370  031050  000200                        RDY                 ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7371  031052  001614                        908.                ;ALLOW 9080 MICRO SECONDS
7372  031054  001507                        839.                ;RDY MUST SET BETWEEN
7373                                                            ;690 AND 17470 MICRO SECONDS
7374
7375                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7376                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
7377
7378  031056  013746  002450                MOV     @#REFOR,-(SP)   ;SAVE COMMAND
7379  031062  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7380  031066  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7381  031072  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7382                                                                ;AND COMMAND SHOULD BE SET
7383  031074  001405                        BEQ     67$             ;BRANCH IF GOOD
7384  031076  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
7385  031102  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
7386  031106  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
7387                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
7388  031110  012746  010500        67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
7389  031114  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7390  031120  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7391                                                                ;SHOULD BE SET
7392  031122  001405                        BEQ     69$             ;BRANCH IF GOOD
7393  031124  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
7394  031130  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
7395  031134  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
7396                                                                ;MOL!DPR!VV SHOULD BE SET
7397  031136                        69$:
7398
7399                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7400
7401  031136  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
7402  031142  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
7403  031144  000000                        0                       ;DATA
7404  031146  004037  041270                JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
7405  031152  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
7406  031154  004544                        REINTO+<260.*2>                 ;DATA
7407  031156  004037  041270                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
7408  031162  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
7409  031164  000002                        2                       ;DATA
7410
7411                                        ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
7412                                        ;*WITH REGISTERS AFTER COMMAND
7413
7414
7415  031166  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7416                                                                ;PRESENT VALUE
7417  031172  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7418  031174  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
```

```
7419  031176  000022                          18.                     ;18. REGISTERS TO BE COMPARED
7420  031200  031204                          5$                      ;RETURN TO 5$ ON ERROR
7421  031202  031210                          6$                      ;RETURN TO 6$ ON NO ERROR
7422
7423
7424  031204  104031              5$:         ERROR   31              ;READ HEADER AND DATA CAUSED
7425  031206  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
7426                                                                  ;GOOD DATA GIVES WHAT SHOULD
7427                                                                  ;BE THERE
7428                                                                  ;RECEIVED DATA GIVES WHAT WAS
7429                                                                  ;THERE AFTER COMMAND
7430
7431                                          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
7432                                          ;*THAT READ WAS GOOD
7433
7434  031210              6$:
7435
7436  031210  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7437  031214  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
7438  031216  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
7439  031220  000404                          260.                    ;260. WORDS TO BE COMPARED
7440  031222  031226                          7$                      ;RETURN TO 7$ ON ERROR
7441  031224  031232                          10$                     ;RETURN TO 10$ ON NO ERROR
7442
7443
7444
7445  031226  104032              7$:         ERROR   32              ;WRITE HEADER AND DATA
7446  031230  000207                          RTS     PC              ;FOLLOWED BY A READ HEADER
7447                                                                  ;AND DATA GAVE A READ ERROR
7448                                                                  ;ERROR MAY BE IN READ OR WRITE
7449  031232              10$:
```

```
                                ;;*****************************************************************
                                ;*TEST 42       IMPLIED SEEK TO CYL 001

                                        .SBTTL  SEEK TESTS

                                ;*      ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM CYLINDER 0
                                ;*      TO CYLINDER 1, A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN
                                ;*      ONE SECTOR OF WORDS
                                ;*
                                ;*
                                ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                                ;*      TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS
                                ;*      (4 HEADER, 256 DATA=!125, 4 HEADER 2 DATA = 2000
                                ;*      THEN READ HEADER AND DATA FOR ABOVE, TRACK 0, SECTOR 0, CYL=1
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      10000,0,0,0 AND 256 OF !125, 10001,0,0,0, AND 2 OF 2000
                                ;*      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 CHANGE
                                ;*      THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
                                ;*
                                ;*
                                ;*      NOW FOR READ COMMAND READ INTO BUFFER IS FILLED
                                ;*      WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH
                                ;*      10001,0,0,0 AND 2 OF 2000,254 OF ZEROS COMMAND IS LOADED
                                ;*      INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN FOR THE READ COMMAND
                                ;*
                                ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                ;*      THEN THE READ DATA IS COMPARED.

                                ;;*****************************************************************
031232  000004                  TST42:  SCOPE
031234  012706  001000                  MOV     #STACK,SP       ;RESET STACK
031240  012737  000042  004604          MOV     #42,@#TSTNM     ;SAVE TEST NUMBER

031246  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*FILL WRITE FROM BUFFER WITH HEADER

031252  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
031256  002470                          WRFROM                  ;LOCATION WHERE SAVED
031260  000004                          4                       ;NUMBER OF WORDS SAVED
031262  010000                          10000                   ;FIRST DATA WORD
031264  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
031266  000000                          0                       ;THIRD DATA WORD
031270  000000                          0                       ;FOURTH DATA WORD
```

```
                                        ;*FILL WRITE FROM BUFFER WITH DATA
031272  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
031276  003500                          WRFROM+10                       ;STARTING FROM WRFROM+10
031300  000400                          256.                    ;256. WORDS
031302  001125                          <18.*40>!21.                    ;FILL WITH <19.*40>!21.


                                        ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER

031304  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
031310  003500                          WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
031312  000004                          4                       ;NUMBER OF WORDS SAVED
031314  010001                          10001                   ;FIRST DATA WORD
031316  000000                          0                       ;SECOND DATA WORD
031320  000000                          0                       ;THIRD DATA WORD
031322  000000                          0                       ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
031324  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
031330  003510                          WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
031332  000002                          2                       ;2 WORDS
031334  002000                          2000                    ;FILL WITH 2000


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


031336  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
031342  003534                          REINTO                  ;LOCATION WHERE SAVED
031344  000004                          4                       ;NUMBER OF WORDS SAVED
031346  010000                          10000                   ;FIRST DATA WORD
031350  011025                          <18.*400!<21.>                  ;SECOND DATA WORD
031352  000000                          0                       ;THIRD DATA WORD
031354  000000                          0                       ;FOURTH DATA WORD
031356  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
031362  003544                          REINTO+10                       ;STARTING FROM REINTO+10
031364  000400                          256.                    ;256. WORDS
031366  001125                          <18.*40>!21.                    ;FILL WITH <19.*40>!21.


031370  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO+<260.*2>
031374  004544                          REINTO+<260.*2>                 ;LOCATION WHERE SAVED
031376  000004                          4                       ;NUMBER OF WORDS SAVED
031400  010001                          10001                   ;FIRST DATA WORD
031402  000000                          0                       ;SECOND DATA WORD
031404  000000                          0                       ;THIRD DATA WORD
031406  000000                          0                       ;FOURTH DATA WORD
031410  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
031414  004554                          REINTO+<264.*2>                 ;STARTING FROM REINTO+<264.*2>
031416  000002                          2                       ;2 WORDS
031420  002000                          2000                    ;FILL WITH 2000
```

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

031422  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
031426  000000                          0                       ;CYLINDER 0
031430      025                 .BYTE   21.                     ;SECTOR 21.
031431      022                 .BYTE   18.                     ;TRACK 18.
031432  177365                          -262.-4                 ;WORD COUNT (DATA) = 262. +
                                                                ;4 HEADER WORDS
031434  002470                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
031436  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
031440  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
031442  002444                          WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1


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

031444  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
031450  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
031452  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
031454  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

031456  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
031462  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
031466  000000                          HALT                    ;STOP

031470  013777  004606  150570          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


031476  013746  002444                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
031502  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
031506  012677  150566                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED
031512  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
031514  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
```

```
7612                                    ;*ONE REVOLUTION = 16670 MICRO1 SEC, ONE SECTOR = 760 MICRO1 SEC
7613
7614
7615  031516  104412                    WAT                     ;WAIT FOR RDY BIT TO SET
7616  031520  002300                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
7617  031522  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7618  031524  003237                    1695.                   ;ALLOW 16950 MICRO SECONDS
7619  031526  001515                    845.                    ;RDY MUST SET BETWEEN
7620                                                            ;8500 AND 25400 MICRO SECONDS
7621
7622                                    ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7623                                    ;*R0 AND R5 IMMEDIATELY AFTER GO
7624
7625  031530  013746  002444            MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
7626  031534  052716  004101            BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7627  031540  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7628  031544  022600                    CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7629                                                            ;AND COMMAND SHOULD BE SET
7630  031546  001405                    BEQ     64$             ;BRANCH IF GOOD
7631  031550  010037  001126            MOV     R0,@#$BDDAT     ;BAD DATA
7632  031554  010137  004600            MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
7633  031560  104021                    ERROR   21              ;DURING ABOVE OPERATION ONLY
7634                                                            ;COMMAND AND IE!GO!DVA SHOULD BE SET
7635  031562  012746  010500    643:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
7636  031566  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7637  031572  022605                    CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7638                                                            ;SHOULD BE SET
7639  031574  001405                    BEQ     66$             ;BRANCH IF GOOD
7640  031576  010537  001126            MOV     R5,@#$BDDAT     ;BAD DATA
7641  031602  010337  004600            MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
7642  031606  104063                    ERROR   63              ;DURING ABOVE OPERATION ONLY
7643                                                            ;MOL!DPR!VV SHOULD BE SET
7644  031610                    66$:
7645
7646                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7647
7648  031610  004037  041270            JSR     R0,@#FILLRE     ;MOV 3 INTO SAVED RHWC
7649  031614  002272                    RHWC                    ;SAVED REGISTER TO CHANGE
7650  031616  000000                    0                       ;DATA
7651  031620  004037  041270            JSR     R0,@#FILLRE     ;MOV WRFROM+(266.*2) INTO SAVED RHBA
7652  031624  002274                    RHBA                    ;SAVED REGISTER TO CHANGE
7653  031626  003514                    WRFROM+(266.*2)                 ;DATA
7654  031630  004037  041270            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCA
7655  031634  002312                    RHCA                    ;SAVED REGISTER TO CHANGE
7656  031636  000001                    1                       ;DATA
7657  031640  004037  041270            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCC
7658  031644  002334                    RHCC                    ;SAVED REGISTER TO CHANGE
7659  031646  000001                    1                       ;DATA
7660  031650  004037  041270            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
7661  031654  002304                    RHDST                   ;SAVED REGISTER TO CHANGE
7662  031656  000001                    1                       ;DATA
7663
7664                                    ;*NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
7665                                    ;*WITH REGISTERS AFTER COMMAND
```

```
                                                    ;COMPARE SAVED REGISTERS WITH
                                                    ;PRESENT VALUE
                                                    ;GOOD DATA SAVED IN 'SAVERE'
                                                    ;TEST DATA STARTING FROM 'F4WC'
                                                    ;18. REGISTERS TO BE COMPARED
                                                    ;RETURN TO 1$ ON ERROR
                                                    ;RETURN TO 2$ ON NO ERROR

                                                    ;WRITE HEADER AND DATA
                                                    ;CAUSED IMPROPER REGISTER
                                                    ;CHANGE
                                                    ;GOOD DATA GIVES WHAT SHOULD
                                                    ;BE THERE
                                                    ;RECEIVED DATA GIVES WHAT
                                                    ;WAS THERE AFTER COMMANT

                                                    ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
                                                    ;*NOTHING GOT CHANGED

                                                    ;COMPARE TWO BLOCKS OF MEMORY
                                                    ;GOOD DATA STARTS FROM REINTO
                                                    ;TEST DATA STARTS FROM WRFROM
                                                    ;266. WORDS TO BE COMPARED
                                                    ;RETURN TO 3$ ON ERROR
                                                    ;RETURN TO 4$ ON NO ERROR

                                                    ;WRITE HEADER AND DATA
                                                    ;CHANGED WRITE FROM BUFFER

                                                    ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
                                                    ;*READ INTO BUFFER IS FILLED WITH ONES

                                                    ;SET R1-RHCS1, R2-RHCS2
                                                    ;R3-RHDS1, R4-RHER1
                                                    ;GIVE RH-11 INITIALIZE
                                                    ;SETUP UNIT NUMBER
                                                    ;CLEAR 260. WORDS, FROM REINTO
                                                    ;STARTING FROM REINTO
                                                    ;260. WORDS
                                                    ;FILL WITH -1

                                                    ;*WRITE FROM BUFFER IS FILLED WITH 10001.0.0.0.2000.2000. AND 254 OF 0

                                                    ;SAVE HEADER DATA IN WRFROM
                                                    ;LOCATION WHERE SAVED
                                                    ;NUMBER OF WORDS SAVED
```

```
7720  031752  010001                          10001                   ;FIRST DATA WORD
7721  031754  000000                          0                       ;SECOND DATA WORD
7722  031756  000000                          0                       ;THIRD DATA WORD
7723  031760  000000                          0                       ;FOURTH DATA WORD
7724  031762  002000                          2000                    ;FIFTH DATA WORD
7725  031764  002000                          2000                    ;SIXTH DATA WORD
7726  031766  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 254. WORDS, FROM WRFROM+(6*2)
7727  031772  002504                          WRFROM+(6*2)            ;STARTING FROM WRFROM+(6*2)
7728  031774  000376                          254.                    ;254. WORDS
7729  031776  000000                          0                       ;FILL WITH 0
7730
7731
7732                                          ;*NOW FILL COMMAND
7733
7734  032000  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7735  032004  000001                          1                       ;CYLINDER 1
7736  032006      000                  .BYTE   0                       ;SECTOR 0
7737  032007      000                  .BYTE   0                       ;TRACK 0
7738  032010  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
7739                                                                  ;4 HEADER WORDS
7740  032012  003534                          REINTO                  ;BUS ADDRESS
7741                                                                  ;STARTING ADDRESS OF DATA
7742                                                                  ;BUFFER = REINTO
7743  032014  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7744  032016  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
7745                                                                  ;INHIBIT ECC CORRECTION
7746                                                                  ;DO NOT INHIBIT HEADER COMPARE
7747  032020  002450                          REFOR                   ;GET READY TO DO A REFOR
7748                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
7749
7750
7751                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7752
7753  032022  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
7754  032026  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7755  032030  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
7756                                                                  ;THE REGISTERS ARE SAVED
7757  032032  000022                          18.                     ;NUMBER OF REGISTERS
7758                                                                  ;SAVED = 18.
7759
7760  032034  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7761                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
7762  03204C  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7763                                                                  ;THE FIRST SET OF BITS DON'T = 1
7764  032044  000000                          HALT                    ;STOP
7765
7766  032046  013777  004606  150212          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7767                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
7768                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7769                                                                  ;'TIME' WILL ONLY SAVE
7770                                                                  ;CURRENT CYLINDER ADDRESS
7771                                                                  ;AND LOOK AHEAD REGISTERS
7772
7773
```

```
 7774  032054  013746  002450          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
 7775  032060  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 7776                                                          ;ENABLE INTERRUPT
 7777  032064  012677  153210          MOV     (SP)+,@RHCS1    ;GO WITH
 7778                                                            ;72 IN RHCS1 FOR READ DATA
 7779                                                          ;WITH INTERRUPT ENABLED
 7780  032070  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 7781  032072  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 7782
 7783
 7784  032074  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
 7785  032076  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
 7786  032100  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 7787  032102  001614                  908.                    ;ALLOW 9090 MICRO SECONDS
 7788  032104  001507                  839.                    ;RDY MUST SET BETWEEN
 7789                                                          ;690 AND 17470 MICRO SECONDS
 7790
 7791                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 7792                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
 7793
 7794  032106  013746  002450          MOV     @#REFOR,-(SP)   ;SAVE COMMAND
 7795  032112  052716  004101          BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
 7796  032116  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7797  032122  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
 7798                                                          ;AND COMMAND SHOULD BE SET
 7799  032124  001405                  BEQ     67$             ;BRANCH IF GOOD
 7800  032126  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
 7801  032132  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 7802  032136  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
 7803                                                          ;COMMAND AND IE!GO!DVA SHOULD BE SET
 7804  032140  012746  010500  67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
 7805  032144  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7806  032150  022605                  CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
 7807                                                          ;SHOULD BE SET
 7808  032152  001405                  BEQ     69$             ;BRANCH IF GOOD
 7809  032154  010537  001126          MOV     R5,@#$BDDAT     ;BAD DATA
 7810  032160  010337  004600          MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 7811  032164  104063                  ERROR   63              ;DURING ABOVE OPERATION ONLY
 7812                                                          ;MOL!DPR!VV SHOULD BE SET
 7813  032166                  69$:
 7814
 7815                                  ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
 7816  032166  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 7817  032172  002272                  RHWC                    ;SAVED REGISTER TO CHANGE
 7818  032174  000000                  0                       ;DATA
 7819  032176  004037  041270          JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
 7820  032202  002274                  RHBA                    ;SAVED REGISTER TO CHANGE
 7821  032204  004544                  REINTO+<260.*2>                 ;DATA
 7822  032206  004037  041270          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
 7823  032212  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
 7824  032214  000001                  1                       ;DATA
 7825
 7826                                  ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
 7827                                  ;*WITH REGISTERS AFTER COMMAND
```

```
7828
7829  032216  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7830                                                                  ;PRESENT VALUE
7831  032222  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7832  032224  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
7833  032226  000022                        18.                     ;18. REGISTERS TO BE COMPARED
7834  032230  032234                        5$                      ;RETURN TO 5$ ON ERROR
7835  032232  032240                        6$                      ;RETURN TO 6$ ON NO ERROR
7836
7837
7838  032234  104031              5$:       ERROR   31              ;READ HEADER AND DATA CAUSED
7839  032236  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
7840                                                                  ;GOOD DATA GIVES WHAT SHOULD
7841                                                                  ;BE THERE
7842                                                                  ;RECEIVED DATA GIVES WHAT WAS
7843                                                                  ;THERE AFTER COMMAND
7844
7845                                        ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
7846                                        ;*THAT READ WAS GOOD
7847
7848  032240                        6$:
7849
7850  032240  004037  043404                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7851  032244  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
7852  032246  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
7853  032250  000404                        260.                    ;260. WORDS TO BE COMPARED
7854  032252  032256                        7$                      ;RETURN TO 7$ ON ERROR
7855  032254  032262                        10$                     ;RETURN TO 10$ ON NO ERROR
7856
7857
7858
7859  032256  104032              7$:       ERROR   32              ;WRITE HEADER AND DATA
7860  032260  000207                        RTS     PC              ;FOLLOWED BY A READ HEADER
7861                                                                  ;AND DATA GAVE A READ ERROR
7862                                                                  ;ERROR MAY BE IN READ OR WRITE
7863  032262                        10$:
7864
```

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

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

                                ;;*****************************************************************
032262  000004                  TST43:  SCOPE
032264  012706  001000                  MOV     #STACK,SP       ;RESET STACK
032270  012737  000043  004604          MOV     #43,@#TSTNM     ;SAVE TEST NUMBER

032276  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

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

032302  012737  010000  032460          MOV     #10000,@#ST1+10
032310  012737  001125  032500          MOV     #<<18.*40>!21.>,@#ST2+10
032316  012737  010001  032512          MOV     #10001,@#ST3+10
032324  012737  002000  032532          MOV     #2000,@#ST4+10
032332  012737  000000  032540          MOV     #0,@#ST5+4
032340  012737  000001  032750          MOV     #1,@#ST6+6
032346  012737  000001  032760          MOV     #1,@#ST6+16
032354  012737  010001  033042          MOV     #10001,@#ST9+10
032362  012737  002000  033062          MOV     #2000,@#ST10+10
032370  012737  000001  033102          MOV     #1,@#ST11+4

                                        ;*THIS IS TO GET THE HEADS TO CYLINDER 0

032376  013777  004606  147662          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

032404  013746  002426                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
```

```
7919  032410  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7920                                                                    ;ENABLE INTERRUPT
7921  032414  012677  147660                  MOV     (SP)+,@RHCS1    ;GO WITH
7922                                                                    ;6 IN RHCS1 FOR RECALIBRATE
7923                                                                    ;WITH INTERRUPT ENABLED
7924  032420  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7925  032422  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7926
7927  032424  104412                          WAT                     ;WAIT FOR DRY BIT TO SET
7928  032426  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
7929  032430  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7930  032432  076377                          31999.                  ;ALLOW 319990 MICRO SECONDS
7931  032434  056701                          24001.                  ;DRY MUST SET BETWEEN
7932                                                                    ;79980 AND 560000 MICRO SECONDS
7933  032436  012737  000012  001200          MOV     #10.,@#STMP1    ;TEN COUNT TO GET TO CYLINDER 10
7934
7935  032444  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7936                                                                    ;R3-RHDS1, R4-RHER1
7937                                                                    ;GIVE RH-11 INITIALIZE
7939                                                                    ;SETUP UNIT NUMBER
7939
7940                                          ;*FILL WRITE FROM BUFFER WITH HEADER
7941  032450                          ST1:
7942
7943  032450  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
7944  032454  002470                          WRFROM                  ;LOCATION WHERE SAVED
7945  032456  000004                          4                       ;NUMBER OF WORDS SAVED
7946  032460  010000                          10000                   ;FIRST DATA WORD
7947  032462  011025                          <18.*400>!21.                   ;SECOND DATA WORD
7948  032464  000000                          0                       ;THIRD DATA WORD
7949  032466  000000                          0                       ;FOURTH DATA WORD
7950
7951                                          ;*FILL WRITE FROM BUFFER WITH DATA
7952  032470                          ST2:
7953  032470  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
7954  032474  002500                          WRFROM+10                       ;STARTING FROM WRFROM+10
7955  032476  000400                          256.                    ;256. WORDS
7956  032500  001125                          <0.*2000>!<18.*40>!21.                  ;FILL WITH <0.*2000>!<18.*40>!21.
7957
7958
7959                                          ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
7960  032502                          ST3:
7961
7962  032502  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
7963  032506  003500                          WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
7964  032510  000004                          4                       ;NUMBER OF WORDS SAVED
7965  032512  010001                          10001                   ;FIRST DATA WORD
7966  032514  000000                          0                       ;SECOND DATA WORD
7967  032516  000000                          0                       ;THIRD DATA WORD
7968  032520  000000                          0                       ;FOURTH DATA WORD
7969
7970                                          ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
7971  032522                          ST4:
7972  032522  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
```

```
7973  032526  003510                          WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
7974  032530  000002                          2                               ;2 WORDS
7975  032532  002000                          1.*2000                         ;FILL WITH 1.*2000
7976
7977
7978                                          ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7979  032534                            ST5:
7980
7981  032534  004037  043340                  JSR     R0,@#RUN                ;SETUP TO RUN FOR DATA COMMAND
7982  032540  000000                          0                               ;CYLINDER 0
7983  032542     025                  .BYTE   21.                             ;SECTOR 21.
7984  032543     022                  .BYTE   18.                             ;TRACK 18.
7985  032544  177366                          -262.-4                         ;WORD COUNT (DATA) = 262. +
7986                                                                          ;4 HEADER WORDS
7987  032546  002470                          WRFROM                          ;BUS ADDRESS
7988                                                                          ;STARTING ADDRESS OF DATA
7989                                                                          ;BUFFER = WRFROM
7990  032550  000000                          0                               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7991  032552  010000                          FMT22                           ;16 BITS PER WORD FORMAT
7992                                                                          ;DO NOT INHIBIT ECC CORRECTION
7993                                                                          ;DO NOT INHIBIT HEADER COMPARE
7994  032554  002444                          WRIFOR                          ;GET READY TO DO A WRIFOR
7995                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7996
7997
7998                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
7999
8000  032556  004037  041534                  JSR     R0,@#SAVER              ;SAVE REGISTERS
8001  032562  002272                          RHWC                            ;RHWC IS THE FIRST REGISTER SAVED
8002  032564  004612                          SAVERE                          ;STARTING ADDRESS OF WHERE
8003                                                                          ;THE REGISTERS ARE SAVED
8004  032566  000022                          18.                             ;NUMBER OF REGISTERS
8005                                                                          ;SAVED = 18.
8006
8007  032570  004737  041446                  JSR     PC,@#CHECKT             ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8008                                                                          ;AND THAT NO STATUS BITS ARE STUCK = 1
8009  032574  104400  065402                  TYPE    .CPHALT                 ;CANNOT CONTINUE TESTING IF ANY OF
8010                                                                          ;THE FIRST SET OF BITS DON'T = 1
8011  032600  000000                          HALT                            ;STOP
8012
8013  032602  013777  004606  147456          MOV     @#RP4VEC,@RPVEC         ;SET RP04 VECTOR ADDRESS
8014                                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
8015                                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8016                                                                          ;'TIME' WILL ONLY SAVE
8017                                                                          ;CURRENT CYLINDER ADDRESS
8018                                                                          ;AND LOOK AHEAD REGISTERS
8019
8020
8021  032610  013746  002444                  MOV     @#WRIFOR,-(SP)          ;GET READY TO MOVE COMMAND
8022  032614  052716  000101                  BIS     #GO!IE,(SP)             ;GET READY TO SET 'GO' AND
8023                                                                          ;ENABLE INTERRUPT
8024  032620  012677  147454                  MOV     (SP)+,@RHCS1            ;GO WITH
8025                                                                           ;62 IN RHCS1 FOR WRITE HEADER AND DATA
8026                                                                          ;WITH INTERRUPT ENABLED
```

```
8027  032624  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8028  032626  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8029
8030                                          ;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760 MICRO SEC.
8031                                          ;*MAX TIME ALLOWED = ONE REVOLUTION + SEEK + 2 SECTORS
8032                                          ;*MIN TIME ALLOWED = 2 SECTORS + SEEK
8033
8034
8035  032630  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
8036  032632  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8037  032634  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8038  032636  003237                          1695.                   ;ALLOW 16950 MICRO SECONDS
8039  032640  001515                          845.                    ;RDY MUST SET BETWEEN
8040                                                                  ;8500 AND 25400 MICRO SECONDS
8041
8042                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8043                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
8044
8045  032642  013746  002444                  MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
8046  032646  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8047  032652  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8048  032656  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8049                                                                  ;AND COMMAND SHOULD BE SET
8050  032660  001405                          BEQ     64$             ;BRANCH IF GOOD
8051  032662  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
8052  032666  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8053  032672  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
8054                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
8055  032674  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8056  032700  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8057  032704  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8058                                                                  ;SHOULD BE SET
8059  032706  001405                          BEQ     66$             ;BRANCH IF GOOD
8060  032710  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
8061  032714  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
8062  032720  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
8063                                                                  ;MOL!DPR!VV SHOULD BE SET
8064  032722                          66$:
8065
8066                                          ;*NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES
8067
8068  032722  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8069  032726  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
8070  032730  000000                          0                       ;DATA
8071  032732  004037  041270                  JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RH3A
8072  032736  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
8073  032740  003514                          WRFROM+<266.*2>                 ;DATA
8074  032742                          ST6:
8075  032742  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCC
8076  032746  002334                          RHCC                    ;SAVED REGISTER TO CHANGE
8077  032750  000001                          1                       ;DATA
8078  032752  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCA
8079  032756  002312                          RHCA                    ;SAVED REGISTER TO CHANGE
8080  032760  000001                          1                       ;DATA
```

```
8081  032762  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8082  032766  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
8083  032770  000001                          1                       ;DATA
8084
8085                                          ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
8086                                          ;*WITH REGISTERS AFTER COMMAND
8087
8088
8089  032772  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8090                                                                  ;PRESENT VALUE
8091  032776  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8092  033000  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
8093  033002  000022                          18.                     ;18. REGISTERS TO BE COMPARED
8094  033004  033010                          ST7                     ;RETURN TO ST7 ON ERROR
8095  033006  033014                          ST8                     ;RETURN TO ST8 ON NO ERROR
8096  033010  104027          ST7:            ERROR   27              ;WRITE HEADER AND DATA CAUSED
8097  033012  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
8098                                                                  ;GOOD DATA GIVES WHAT SHOULD BE
8099                                                                  ;THERE
8100                                                                  ;RECEIVED DATA GIVES WHAT WAS BE
8101                                                                  ;THERE AFTER COMMAND
8102
8103                                          ;*SETUP TO READ HEADER AND DATA FOR NEXT TRACK
8104                                          ;*FILL READ INTO BUFFER WITH ALL ONES
8105
8106  033014          ST8:
8107
8108  033014  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8109                                                                  ;R3-RHDS1, R4-RHER1
8110                                                                  ;GIVE RH-11 INITIALIZE
8111                                                                  ;SETUP UNIT NUMBER
8112  033020  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
8113  033024  003534                          REINTO                  ;STARTING FROM REINTO
8114  033026  000404                          260.                    ;260. WORDS
8115  033030  177777                          -1                      ;FILL WITH -1
8116
8117
8118                                          ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
8119
8120  033032          ST9:
8121
8122  033032  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8123  033036  002470                          WRFROM                  ;LOCATION WHERE SAVED
8124  033040  000004                          4                       ;NUMBER OF WORDS SAVED
8125  033042  010001                          10001                   ;FIRST DATA WORD
8126  033044  000000                          0                       ;SECOND DATA WORD
8127  033046  000000                          0                       ;THIRD DATA WORD
8128  033050  000000                          0                       ;FOURTH DATA WORD
8129  033052          ST10:
8130  033052  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<4*2>
8131  033056  002500                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
8132  033060  000002                          2                       ;2 WORDS
8133  033062  002000                          1*2000                  ;FILL WITH 1*2000
8134
```

```
8135  033064  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
8136  033070  002504                          WRFROM+<6*2>            ;STARTING FROM WRFROM+<6*2>
8137  033072  000376                          254.                    ;254. WORDS
8138  033074  000000                          0                       ;FILL WITH 0
8139
8140
8141                                          ;*FILL COMMAND INTO REGISTERS
8142  033076                          ST11:
8143
8144  033076  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
8145  033102  000001                          1                       ;CYLINDER 1
8146  033104     000                  .BYTE   0                       ;SECTOR 0
8147  033105     000                  .BYTE   0                       ;TRACK 0
8148  033106  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
8149                                                                  ;4 HEADER WORDS
8150  033110  003534                          REINTO                  ;BUS ADDRESS
8151                                                                  ;STARTING ADDRESS OF DATA
8152                                                                  ;BUFFER = REINTO
8153  033112  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8154  033114  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
8155                                                                  ;INHIBIT ECC CORRECTION
8156                                                                  ;DO NOT INHIBIT HEADER COMPARE
8157  033116  002450                          REFOR                   ;GET READY TO DO A REFOR
8158                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
8159
8160
8161                                          ;*SAVE REGISTERS FOR COMPARISON AFTER READ HEADER
8162                                          ;*AND DATA
8163
8164  033120  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
8165  033124  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
8166  033126  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
8167                                                                  ;THE REGISTERS ARE SAVED
8168  033130  000020                          16.                     ;NUMBER OF REGISTERS
8169                                                                  ;SAVED = 16.
8170
8171  033132  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8172                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
8173  033136  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
8174                                                                  ;THE FIRST SET OF BITS DON'T = 1
8175  033142  000000                          HALT                    ;STOP
8176
8177  033144  013777  004606  147114          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8178                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
8179                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8180                                                                  ;'TIME' WILL ONLY SAVE
8181                                                                  ;CURRENT CYLINDER ADDRESS
8182                                                                  ;AND LOOK AHEAD REGISTERS
8183
8184
8185  033152  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
8186  033156  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
8187                                                                  ;ENABLE INTERRUPT
8188  033162  012677  147112                  MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED
033166  011100                    MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
033170  011305                    MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION


033172  104412                    WAT                     ;WAIT FOR RDY BIT TO SET
033174  002300                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
033176  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
033200  001614                    908.                    ;ALLOW 9080 MICRO SECONDS
033202  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

033204  013746  002450            MOV     @#REFOR,-(SP)   ;SAVE COMMAND
033210  052716  004101            BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
033214  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
033220  022600                    CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                        ;AND COMMAND SHOULD BE SET
033222  001405                    BEQ     64$             ;BRANCH IF GOOD
033224  010037  001126            MOV     R0,@#$BDDAT     ;BAD DATA
033230  010137  004600            MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
033234  104021                    ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                        ;COMMAND AND IE!GO!DVA SHOULD BE SET
033236  012746  010500    64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
033242  011637  001124            MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
033246  022605                    CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
033250  001405                    BEQ     66$             ;BRANCH IF GOOD
033252  010537  001126            MOV     R5,@#$BDDAT     ;BAD DATA
033256  010337  004600            MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
033262  104063                    ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
033264                  66$:

                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

033264  004037  041270            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
033270  002272                    RHWC                    ;SAVED REGISTER TO CHANGE
033272  000000                    0                       ;DATA
033274  004037  041270            JSR     R0,@#FILLRE     ;MOV REINTO+(260.*2) INTO SAVED RHBA
033300  002274                    RHBA                    ;SAVED REGISTER TO CHANGE
033302  004544                    REINTO+(260.*2)                 ;DATA
033304  004037  041270            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
033310  002304                    RHDST                   ;SAVED REGISTER TO CHANGE
033312  000001                    1                       ;DATA

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

033314  004037  042354            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
                                                        ;PRESENT VALUE
033320  004612                          SAVERE          ;GOOD DATA SAVED IN 'SAVERE'
033322  002354                          WC              ;TEST DATA STARTING FROM 'RHWC'
033324  000022                          18.             ;18. REGISTERS TO BE COMPARED
033326  033332                          ST12            ;RETURN TO ST12 ON ERROR
033330  033336                          ST13            ;RETURN TO ST13 ON NO ERROR

033332  104031                  ST12:   ERROR   31      ;READ HEADER AND DATA CAUSED
033334  000207                          RTS     PC      ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER COMMAND

                                        ;*READ INTO BUFFER IS CHECKED FOR PROPER READ
033336                          ST13:

033336  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
033342  002470                          WRFROM          ;GOOD DATA STARTS FROM WRFROM
033344  003534                          REINTO          ;TEST DATA STARTS FROM REINTO
033346  000404                          260.            ;260. WORDS TO BE COMPARED
033350  033354                          ST14            ;RETURN TO ST14 ON ERROR
033352  033360                          ST15            ;RETURN TO ST15 ON NO ERROR


033354  104032                  ST14:   ERROR   32      ;WRITE HEADER AND DATA
033356  000207                          RTS     PC      ;WITH AN IMPLIED SEEK
                                                        ;FOLLOWED BY A READ
                                                        ;HEADER AND DATA ON THE
                                                        ;NEXT TRACK GAVE A
                                                        ;READ ERROR
                                                        ;ERROR MAY BE READ OR WRITE

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

033360  005237  032460          ST15:   INC     @#ST1+10
033364  062737  002000  032500          ADD     #<1.*2000>,@#ST2+10
033372  005237  032512                  INC     @#ST3+10
033376  062737  002000  032532          ADD     #<1.*2000>,@#ST4+10
033404  005237  032540                  INC     @#ST5+4
033410  005237  032750                  INC     @#ST6+6
033414  005237  032760                  INC     @#ST6+16
033420  005237  033042                  INC     @#ST9+10
033424  062737  002000  033062          ADD     #<1.*2000>,@#ST10+10
033432  005237  033102                  INC     @#ST11+4
033436  005337  001200                  DEC     @#$TMP1         ;COUNT FOR TEN TIMES
033442  001001                          BNE     ST16            ;BRANCH IF 10 NOT DONE
033444  000402                          BR      ST17            ;10 COMPLETED SO CONTINUE
033446  000137  032450          ST16:   JMP     ST1             ;JUMP AS 10 NOT DONE
```

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

033452                          ST17:

033452  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER

033456  004037  041336                JSR     R0,@#SEEKCY     ;SEEK FOR
033462  000000                        0                       ;CYLINDER 0

                                        ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
033464  004037  041534                JSR     R0,@#SAVER      ;SAVE REGISTERS
033470  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
033472  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                              ;THE REGISTERS ARE SAVED
033474  000022                        18.                     ;NUMBER OF REGISTERS
                                                              ;SAVED = 18.

033476  013777  004636  146562        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


033504  013746  002452                MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
033510  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                              ;ENABLE INTERRUPT
033514  012677  146560                MOV     (SP)+,@RHCS1    ;GO WITH
                                                               ;4 IN RHCS1 FOR SEEK
                                                              ;WITH INTERRUPT ENABLED
033520  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
033522  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

                                        ;*SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SEC

033524  104412                        WAT                     ;WAIT FOR DRY BIT TO SET
033526  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
033530  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
033532  015530                        7000.                   ;ALLOW 70000 MICRO SECONDS
033534  000043                        35.                     ;DRY MUST SET BETWEEN
                                                              ;69650 AND 70350 MICRO SECONDS

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

033536  013746  002452                MOV     @#SEECOM,-(SP)  ;SAVE COMMAND
033542  052716  004301                BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
```

```
 8351  033546  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 8352  033552  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY D/A!GO!IE!RDY
 8353                                                                 ;AND COMMAND SHOULD BE SET
 8354  033554  001405                        BEQ     64$             ;BRANCH IF GOOD
 8355  033556  010037  001136                MOV     R0,@#$BDDAT     ;BAD DATA
 8356  033562  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 8357  033566  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
 8358                                                                 ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
 8359  033570  012746  030500        64$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHCS1
 8360  033574  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 8361  033600  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
 8362                                                                 ;SHOULD BE SET
 8363  033602  001405                        BEQ     66$             ;BRANCH IF GOOD
 8364  033604  010537  001136                MOV     R5,@#$BDDAT     ;BAD DATA
 8365  033610  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 8366  033614  104063                        ERROR   53              ;DURING ABOVE OPERATION ONLY
 8367                                                                 ;PIP!MOL!DPR!VV SHOULD BE SET
 8368  033616                        66$:
 8369
 8370                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
 8371
 8372  033616  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
 8373  033622  002334                        RHCC                    ;SAVED REGISTER TO CHANGE
 8374  033624  000000                        0                       ;DATA
 8375
 8376  033626  053737  004740  004636        BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
 8377                                                                 ;FOR WORKING DRIVE IN
 8378                                                                 ;SAVED RHAS LOACTION
 8379
 8380  033634  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 8381  033640  002322                        RHDS1                   ;CHANGE RHDS1 REGISTER
 8382
 8383  033642  000001                        1                       ;1 BIT/BITS TO BE CHANGED
 8384  033644  000001                        1                       ;NEW VALUE OF ATA IS 1
 8385  033646  100000                        ATA                     ;CHANGE ATA BIT
 8386
 8387  033650  004037  042246                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 8388  033654  002300                        RHCS1                   ;CHANGE RHCS1 REGISTER
 8389
 8390  033656  000001                        1                       ;1 BIT/BITS TO BE CHANGED
 8391  033660  000001                        1                       ;NEW VALUE OF SC IS 1
 8392  033662  100000                        SC                      ;CHANGE SC BIT
 8393
 8394                                        ;*COMPARE REGISTERS AFTER A SEEK COMMAND
 8395
 8396  033664  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 8397                                                                 ;PRESENT VALUE
 8398  033670  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 8399  033672  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 8400  033674  000022                        18.                     ;18. REGISTERS TO BE COMPARED
 8401  033676  033702                        ST18                    ;RETURN TO ST18 ON ERROR
 8402  033700  033706                        ST19                    ;RETURN TO ST19 ON NO ERROR
 8403
 8404  033702  104037          ST18:   ERROR   37              ;SEEK COMMAND CAUSED AN
```

```
                                        RTS     PC              ;ERROR
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER A SEEK COMMAND

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

033706                          ST19:
033706  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
033712  003534                          REINTO                  ;STARTING FROM REINTO
033714  000404                          260.                    ;260. WORDS
033716  177777                          -1                      ;FILL WITH -1

                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

033720  004037  041212                  JSR     RC,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
033724  002470                          WRFROM                  ;LOCATION WHERE SAVED
033726  000004                          4                       ;NUMBER OF WORDS SAVED
033730  010000                          10000                   ;FIRST DATA WORD
033732  011025                          (18.*400)!(21.)                 ;SECOND DATA WORD
033734  000000                          0                       ;THIRD DATA WORD
033736  000000                          0                       ;FOURTH DATA WORD
033740  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(4.*2)
033744  002500                          WRFROM+(4.*2)                   ;STARTING FROM WRFROM+(4.*2)
033746  000012                          10.                     ;10. WORDS
033750  001125                          (18.*40)!(21.)                  ;FILL WITH (18.*40)!(21.)

033752  004037  041235                  JSR     R0,@#CLAREA     ;CLEAR 246. WORDS, FROM WRFROM+(14.*2)
033756  002524                          WRFROM+(14.*2)                  ;STARTING FROM WRFROM+(14.*2)
033760  000366                          246.                    ;246. WORDS
033762  177777                          -1                      ;FILL WITH -1


                                        ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS

033764  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
033770  000000                          0                       ;CYLINDER 0
033772      025                 .BYTE   21.                     ;SECTOR 21.
033773      022                 .BYTE   18.                     ;TRACK 18.
033774  177762                          -10.-4                  ;WORD COUNT (DATA) = 10. +
                                                                ;4 HEADER WORDS
033776  003534                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
034000  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
034002  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
```

```
                                                        ;DO NOT INHIBIT HEADER COMPARE
034004  002450                          REFOR           ;GET READY TO DO A REFOR
                                                        ;READ HEADER AND DATA WITH 72 IN RHCS1


034006  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
034012  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
034016  000000                          HALT            ;STOP

034020  013777  004636  146240          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


034026  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
034032  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
034036  012677  146236                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                          ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED


034042  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
034044  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
034046  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
034050  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
034052  001507                          839.                    ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS


                                        ;*CHECK READ WORDS

034054  004037  043404                  JSR     RO,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
034060  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
034062  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
034064  000404                          260.                    ;260. WORDS TO BE COMPARED
034066  034072                          ST26                    ;RETURN TO ST26 ON ERROR
034070  034076                          ST27                    ;RETURN TO ST27 ON NO ERROR


034072  104032          ST26:           ERROR   32              ;READ HEADER AND DATA
034074  000207                          RTS     PC              ;FOLLOWING A SEEK TO CYLINDER 0
                                                        ;FROM CYLINDER 10 GAVE AN
                                                        ;ERROR


034076          ST27:
```

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

                                ;*      THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE
                                ;*      PREVIOUS TEST.  AT THIS POINT THE HEADS ARE ON
                                ;*      CYLINDER 0
                                ;*
                                ;*      ALL REGISTERS ARE SAVED
                                ;*      A SEEK IS GIVEN TO CYLINDER 9
                                ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                ;*      DATA WRITTEN ON CYLINDER 9 IS CHECKED
                                ;;*****************************************************************
034076  000004                 TST44:  SCOPE
034100  012706  001000                 MOV     #STACK,SP       ;RESET STACK
034104  012737  000044  004604         MOV     #44,@#TSTNM     ;SAVE TEST NUMBER

034112  004737  041366                 JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*THIS GETS HEADS TO CYLINDER 0
034116  013777  004606  146142         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

034124  013746  002426                 MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
034130  052716  000101                 BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
034134  012677  146140                 MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ;5 IN RHCS1 FOR RECALIBRATE
                                                                ;WITH INTERRUPT ENABLED
034140  011100                         MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
034142  011305                         MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

034144  104412                         WAT                     ;WAIT FOR DRY BIT TO SET
034146  002322                         RHDS1                   ;WAIT FOR RHDS1 REGISTER
034150  000200                         DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
034152  076377                         31999.                  ;ALLOW 319990 MICRO SECONDS
034154  056701                         24001.                  ;DRY MUST SET BETWEEN
                                                                ;79980 AND 560000 MICRO SECONDS

                                ;*FILL REGISTERS FOR A SEEK COMMAND

034156  004037  041336                 JSR     R0,@#SEEKCY     ;SEEK FOR
034162  000012                         10.                     ;CYLINDER 10.
```

```
                                        ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
034164  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
034170  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
034172  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
034174  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

034176  013777  004606  146062          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

034204  013746  002452                  MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
034210  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
034214  012677  146060                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;4 IN RHCS1 FOR SEEK
                                                                ;WITH INTERRUPT ENABLED
034220  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
034222  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

                                        ;*SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC

034224  104412                          WAT                     ;WAIT FOR DRY BIT TO SET
034226  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
034230  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
034232  015530                          7000.                   ;ALLOW 70000 MICRO SECONDS
034234  000043                          35.                     ;DRY MUST SET BETWEEN
                                                                ;69650 AND 70350 MICRO SECONDS

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

034236  013746  002452                  MOV     @#SEECOM,-(SP)  ;SAVE COMMAND
034242  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
034246  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
034252  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
                                                                ;AND COMMAND SHOULD BE SET
034254  001405                          BEQ     67$             ;BRANCH IF GOOD
034256  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
034262  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
034266  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
034270  012746  030500          67$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
034274  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
034300  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
                                                                ;SHOULD BE SET
034302  001405                          BEQ     69$             ;BRANCH IF GOOD
034304  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
034310  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
```

```
8618  034314  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
8619                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
8620  034316                          69$:
8621
8622                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
8623
8624  034316  004037  041270                  JSR     R0,@#FILLRE     ;MOV 10. INTO SAVED RHCC
8625  034322  002334                          RHCC                    ;SAVED REGISTER TO CHANGE
8626  034324  000012                          10.                     ;DATA
8627
8628  034326  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
8629                                                                  ;FOR WORKING DRIVE IN
8630                                                                  ;SAVED RHAS LOACTION
8631
8632  034334  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
8633  034340  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
8634
8635  034342  000001                          1                       ;1 BIT/BITS TO BE CHANGED
8636  034344  000001                          1                       ;NEW VALUE OF ATA IS 1
8637  034346  100000                          ATA                     ;CHANGE ATA BIT
8638
8639  034350  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
8640  034354  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
8641
8642  034356  000001                          1                       ;1 BIT/BITS TO BE CHANGED
8643  034360  000001                          1                       ;NEW VALUE OF SC IS 1
8644  034362  100000                          SC                      ;CHANGE SC BIT
8645
8646                                          ;*COMPARE REGISTERS AFTER A SEEK COMMAND
8647
8648  034364  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8649                                                                  ;PRESENT VALUE
8650  034370  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8651  034372  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
8652  034374  000022                          18.                     ;18. REGISTERS TO BE COMPARED
8653  034376  034402                          1$                      ;RETURN TO 1$ ON ERROR
8654  034400  034406                          2$                      ;RETURN TO 2$ ON NO ERROR
8655
8656  034402  104037                  1$:     ERROR   37              ;SEEK COMMAND CAUSED
8657  034404  000207                          RTS     PC              ;ERROR
8658                                                                  ;GOOD DATA GIVES WHAT SHOULD
8659                                                                  ;BE THERE
8660                                                                  ;RECEIVED DATA GIVES WHAT WAS
8661                                                                  ;THERE AFTER A SEEK COMMAND
8662
8663                                          ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
8664                                          ;*NO ERROR BITS
8665                                          ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
8666                                          ;*SECTOR 21 TRACK 18. EXPECTED DATA IS 23125
8667                                          ;*FOR 20 WORDS
8668                                          ;*CLEAR READ INTO BUFFER WITH ALL ONES
8669
8670  034406                          2$:
8671
```

```
8672  034406  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8673                                                                  ;R3-RHDS1, R4-RHER1
8674                                                                  ;GIVE RH-11 INITIALIZE
8675                                                                  ;SETUP UNIT NUMBER
8676  034412  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
8677  034416  003534                          REINTO                  ;STARTING FROM REINTO
8678  034420  000404                          260.                    ;260. WORDS
8679  034422  177777                          -1                      ;FILL WITH -1
8680
8681
8682                                          ;*FILL  WRITE FROM BUFFER WITH EXPECTED DATA
8683
8684
8685  034424  004037  041212                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8686  034430  002470                          WRFROM                  ;LOCATION WHERE SAVED
8687  034432  000004                          4                       ;NUMBER OF WORDS SAVED
8688  034434  010011                          10011                   ;FIRST DATA WORD
8689  034436  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
8690  034440  000000                          0                       ;THIRD DATA WORD
8691  034442  000000                          0                       ;FOURTH DATA WORD
8692  034444  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 20. WORDS, FROM WRFROM+<4.*2>
8693  034450  002500                          WRFROM+<4.*2>                   ;STARTING FROM WRFROM+<4.*2>
8694  034452  000024                          20.                     ;20. WORDS
8695  034454  023125                          <9.*2000>!<18.*40>!<21.>                        ;FILL WITH <9.*2000>!<18.*40>!<2
8696
8697  034456  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 250. WORDS, FROM WRFROM+<24.*2>
8698  034462  002550                          WRFROM+<24.*2>                  ;STARTING FROM WRFROM+<24.*2>
8699  034464  000372                          250.                    ;250. WORDS
8700  034466  177777                          -1                      ;FILL WITH -1
8701
8702
8703                                          ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
8704
8705  034470  004037  043343                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
8706  034474  000011                          9.                      ;CYLINDER 9.
8707  034476     025                  .BYTE   21.                     ;SECTOR 21.
8708  034477     022                  .BYTE   18.                     ;TRACK 18.
8709  034500  177750                          -20.-4                  ;WORD COUNT (DATA) = 20. +
8710                                                                  ;4 HEADER WORDS
8711  034502  003534                          REINTO                  ;BUS ADDRESS
8712                                                                  ;STARTING ADDRESS OF DATA
8713                                                                  ;BUFFER = REINTO
8714  034504  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8715  034506  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
8716                                                                  ;INHIBIT ECC CORRECTION
8717                                                                  ;DO NOT INHIBIT HEADER COMPARE
8718  034510  002450                          REFOR                   ;GET READY TO DO A REFOR
8719                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
8720
8721
8722  034512  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8723                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
8724  034516  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
8725                                                                  ;THE FIRST SET OF BITS DON'T = 1
```

```
8726  034522  000000                          HALT                    ;STOP
8727
8728  034524  013777  004606  145534          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8729                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
8730                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8731                                                                  ;'TIME' WILL ONLY SAVE
8732                                                                  ;CURRENT CYLINDER ADDRESS
8733                                                                  ;AND LOOK AHEAD REGISTERS
8734
8735
8736  034532  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
8737  034536  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
8738                                                                  ;ENABLE INTERRUPT
8739  034542  012677  145532                  MOV     (SP)+,@RHCS1    ;GO WITH
8740                                                                    ;72 IN RHCS1 FOR READ DATA
8741                                                                  ;WITH INTERRUPT ENABLED
8742
8743
8744  034546  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
8745  034550  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8746  034552  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8747  034554  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
8748  034556  001507                          839.                    ;RDY MUST SET BETWEEN
8749                                                                  ;690 AND 17470 MICRO SECONDS
8750
8751                                          ;*CHECK READ WORDS
8752
8753  034560  004037  043404                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
8754  034564  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
8755  034566  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
8756  034570  000404                          250.                    ;250. WORDS TO BE COMPARED
8757  034572  034576                          3$                      ;RETURN TO 3$ ON ERROR
8758  034574  034602                          4$                      ;RETURN TO 4$ ON NO ERROR
8759
8760
8761  034576  104032                  3$:     ERROR   32              ;READ HEADER AND DATA
8762  034600  000207                          RTS     PC              ;FOLLOWING A SEEK TO CYLINDER 9
8763                                                                  ;FROM CYLINDER 0 GAVE AN
8764                                                                  ;ERROR
8765  034602                          4$:
8766
8767
8768
8769
```

```
 8770                                   .SBTTL  WRITE CHECK DATA & WRITE PROTECT TESTS
 8771
```

```
                                        ;;**********************************************************
                                        ;*TEST 45       WRITE CHECK HEADER AND DATA
                                        ;
                                        ;*      WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PER WORD
                                        ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266
                                        ;*      CONSISTING OF
                                        ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                                        ;*      10 WORDS OF 177777
                                        ;*      10 WORDS OF 0
                                        ;*      10 WORDS OF 052525
                                        ;*      10 WORDS OF 125252
                                        ;*      16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)
                                        ;*      16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                                        ;*      174 WORDS OF 377
                                        ;*      4 WORDS OF HEADER
                                        ;*      2 WORDS OF 12345
                                        ;*
                                        ;*
                                        ;*      FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND
                                        ;*      DATA COMMAND
                                        ;*      CHECK FOR NO ERRORS
                                        ;*      THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA
                                        ;*      CHECK FOR NO ERRORS
                                        ;*      THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN
                                        ;;**********************************************************
034602  000004                          TST45:  SCOPE
034604  012706  001000                          MOV     #STACK,SP       ;RESET STACK
034610  012737  000045  004604                  MOV     #45,@#TSTNM     ;SAVE TEST NUMBER
                                        
034616  004737  041366                          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
034622  004737  041446                          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
034626  104400  066402                          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                        ;THE FIRST SET OF BITS DON'T = 1
034632  000000                          HALT                            ;STOP
                                        
                                        ;*GET HEADS TO CYLINDER 5
                                        
034634  004037  041336                          JSR     R0,@#SEEKCY     ;SEEK FOR
034640  000005                          5                               ;CYLINDER 5
                                        
034642  013777  004606  145416                  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
                                        
034650  013746  002452                          MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
034654  052716  000101                          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
```

```
                                                        ;ENABLE INTERRUPT
034660  012677  145414          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;4 IN RHCS1 FOR SEEK
                                                        ;WITH INTERRUPT ENABLED
034664  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
034666  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

034670  104412                  WAT                     ;WAIT FOR DRY BIT TO SET
034672  003328                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
034674  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
034676  015530                  70000.                  ;ALLOW 70000 MICRO SECONDS
034700  000043                  35.                     ;DRY MUST SET BETWEEN
                                                        ;69650 AND 70350 MICRO SECONDS

034702  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                                ;*FILL WRITE FROM BUFFER WITH HEADER

034706  004037  041212          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
034712  002470                  WRFROM                  ;LOCATION WHERE SAVED
034714  000004                  4                       ;NUMBER OF WORDS SAVED
034716  010005                  10005                   ;FIRST DATA WORD
034720  003404                  (7*400)!4                       ;SECOND DATA WORD
034722  000000                  0                       ;THIRD DATA WORD
034724  000000                  0                       ;FOURTH DATA WORD

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

034726  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(4*2)
034732  002500                  WRFROM+(4*2)                    ;STARTING FROM WRFROM+(4*2)
034734  000012                  10.                     ;10. WORDS
034736  012344                  (5*2000)!(7*40)!4                       ;FILL WITH (5*2000)!(7*40)!4

034740  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(14.*2)
034744  002524                  WRFROM+(14.*2)                  ;STARTING FROM WRFROM+(14.*2)
034746  000012                  10.                     ;10. WORDS
034750  177777                  -1                      ;FILL WITH -1

034752  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(24.*2)
034756  002550                  WRFROM+(24.*2)                  ;STARTING FROM WRFROM+(24.*2)
034760  000012                  10.                     ;10. WORDS
034762  000000                  0                       ;FILL WITH 0

034764  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(34.*2)
034770  002574                  WRFROM+(34.*2)                  ;STARTING FROM WRFROM+(34.*2)
034772  000012                  10.                     ;10. WORDS
034774  052525                  52525                   ;FILL WITH 52525

034776  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+(44.*2)
```

```
 8980  035002  002620                        WRFROM+<44.*2>                  ;STARTING FROM WRFROM+<44.*2>
 8981  035004  000012                        10.                     ;10. WORDS
 8982  035006  125252                        125252                  ;FILL WITH 125252
 8983
 8984
 8985                                        ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>
 8986
 8987  035010  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8988  035012  012700  177776                MOV     #177776,R0      ;DATA
 8989  035016  012705  000020                MOV     #16.,R5         ;COUNT
 8990  035022  012701  002644                MOV     #WRFROM+<54.*2>,R1 ;WHERE DATA GOES
 8991  035026  000261                        SEC
 8992  035030  010021              1$:       MOV     R0,(R1)+        ;STORE DATA
 8993  035032  006100                        ROL     R0              ;GET ZERO ONE BIT LEFT
 8994  035034  005305                        DEC     R5              ;COUNT 16
 8995  035036  001374                        BNE     1$              ;BRANCH IF 16 NOT DONE
 8996
 8997                                        ;*FILL LEFT ROTATING ONE INTO WRFROM+ 65.*2>
 8998
 8999  035040  000241                        CLC
 9000  035042  012700  000001                MOV     #1,R0
 9001  035046  010021              2$:       MOV     R0,(R1)+
 9002  035050  006300                        ASL     R0
 9003  035052  103375                        BCC     2$
 9004  035054  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9005
 9006
 9007                                        ;*FILL REST OF DATA
 9008
 9009  035056  004037  041236                JSR     R0,3*CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+<56.*2>
 9010  035062  002744                        WRFROM+<86.*2>                  ;STARTING FROM WRFROM+<86.*2>
 9011  035064  000256                        174.                    ;174. WORDS
 9012  035066  000377                        377                     ;FILL WITH 377
 9013
 9014
 9015  035070  004037  041212                JSR     R0,3*FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
 9016  035074  003500                        WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
 9017  035076  000004                        4                       ;NUMBER OF WORDS SAVED
 9018  035100  010005                        10005                   ;FIRST DATA WORD
 9019  035102  003405                        <7*400>!5                       ;SECOND DATA WORD
 9020  035104  000000                        0                       ;THIRD DATA WORD
 9021  035106  000000                        0                       ;FOURTH DATA WORD
 9022  035110  004037  041236                JSR     R0,3*CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
 9023  035114  003510                        WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
 9024  035116  000002                        2                       ;2 WORDS
 9025  035120  012345                        <5*2000>!<7*40>!5               ;FILL WITH <5*2000>!<7*40>!5
 9026
 9027
 9028
 9029                                        ;*READ INTO BUFFER WILL BE CLEARED
 9030  035122  004037  041236                JSR     R0,3*CLAREA     ;CLEAR 266. WORDS, FROM REINTO
 9031  035126  003534                        REINTO                  ;STARTING FROM REINTO
 9032  035130  000412                        266.                    ;266. WORDS
 9033  035132  177400                        177400                  ;FILL WITH 177400
```

```
                                        ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED

035134  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
035140  000005                          5                       ;CYLINDER 5
035142      004                 .BYTE   4                       ;SECTOR 4
035143      007                 .BYTE   7                       ;TRACK 7
035144  177366                          -262.-4                 ;WORD COUNT (DATA) = 262. +
                                                                ;4 HEADER WORDS
035146  002470                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
035150  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
035152  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
035154  002444                          WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1

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

035156  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
035162  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
035164  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
035166  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

035170  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MCL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
035174  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
035200  000000                          HALT                    ;STOP

035202  013777  004606  145056          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 LOCK AHEAD REGISTERS

035210  013746  002444                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
035214  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
035220  012677  145054                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                ; 62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED

035224  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
035226  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
                                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
                                        986.                    ;ALLOW 9860 MICRO SECONDS
                                        934.                    ;RDY MUST SET BETWEEN
                                                                ;1520 AND 18200 MICRO SECONDS


                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
                                        JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
                                        RHWC                    ;SAVED REGISTER TO CHANGE
                                        0                       ;DATA
                                        JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
                                        RHBA                    ;SAVED REGISTER TO CHANGE
                                        WRFROM+<266.*2>                 ;DATA
                                        JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
                                        RHDST                   ;SAVED REGISTER TO CHANGE
                                        3406                    ;DATA

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


                                        JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
                                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
                                        WC                      ;TEST DATA STARTING FROM 'RHWC'
                                        18.                     ;18. REGISTERS TO BE COMPARED
                                        3$                      ;RETURN TO 3$ ON ERROR
                                        4$                      ;RETURN TO 4$ ON NO ERROR


                                3$:     ERROR   27              ;WRITE HEADER AND DATA
                                        RTS     PC              ;CAUSED IMPROPER REGISTER
                                                                ;CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER COMMANT


                                        ;*NOW FILL COMMAND FOR READ

                                4$:

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


                                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
                                        5                       ;CYLINDER 5
                                .BYTE   4                       ;SECTOR 4
                                .BYTE   7                       ;TRACK 7
                                        -262.-4                 ;WORD COUNT (DATA) = 262. +
                                                                ;4 HEADER WORDS
                                        REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
```

```
035330  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
035332  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
035334  002450                          REFOR                   ;GET READY TO DO A REFOR
                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1


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

035336  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
035342  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
035344  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
035346  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

035350  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
035354  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
035360  000000                          HALT                    ;STOP

035362  013777  004636  144676          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


035370  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
035374  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
035400  012677  144674                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                  ;72 IN RHCS1 FOR READ DATA
                                                                ;WITH INTERRUPT ENABLED


035404  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
035406  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
035410  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
035412  001732                          986.                    ;ALLOW 9860 MICRO SECONDS
035414  001502                          834.                    ;RDY MUST SET BETWEEN
                                                                ;1520 AND 18200 MICRO SECONDS


                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

035416  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
035422  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
035424  000000                          0                       ;DATA
035426  004037  041270                  JSR     R0,@#FILLRE     ;MOV REINTO+<266.*2> INTO SAVED RHBA
035432  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
035434  004560                          REINTO+<266.*2>                 ;DATA
```

```
 9096  035436  004037  041270                JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
 9097  035442  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
 9098  035444  003406                        3406                    ;DATA
 9099
 9100                                        ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
 9101                                        ;*WITH REGISTERS AFTER COMMAND
 9102
 9103
 9104
 9105  035446  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 9106                                                                  ;PRESENT VALUE
 9107  035452  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 9108  035454  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 9109  035456  000022                        18.                     ;18. REGISTERS TO BE COMPARED
 9110  035460  035464                        5$                      ;RETURN TO 5$ ON ERROR
 9111  035462  035470                        6$                      ;RETURN TO 6$ ON NO ERROR
 9112
 9113
 9114  035464  104031              5$:       ERROR   31              ;READ HEADER AND DATA CAUSED
 9115  035466  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
 9116                                                                  ;GOOD DATA GIVES WHAT SHOULD
 9117                                                                  ;BE THERE RECEIVED DATA GIVES WHAT WAS
 9118                                                                  ;RECEIVED DATA GIVES WHAT WAS
 9119                                                                  ;THERE AFTER COMMAND
 9120
 9121                                        ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
 9122                                        ;*THAT READ WAS GOOD
 9123
 9124  035470                    6$:
 9125
 9126  035470  004037  043404                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 9127  035474  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 9128  035476  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
 9129  035500  000412                        266.                    ;266. WORDS TO BE COMPARED
 9130  035502  035506                        7$                      ;RETURN TO 7$ ON ERROR
 9131  035504  035512                        10$                     ;RETURN TO 10$ ON NO ERROR
 9132
 9133
 9134
 9135  035506  104032              7$:       ERROR   32              ;WRITE HEADER AND DATA
 9136  035510  000207                        RTS     PC              ;FOLLOWED BY A READ HEADER
 9137                                                                  ;AND DATA GAVE A READ ERROR
 9138                                                                  ;ERROR MAY BE IN READ OR WRITE
 9139
 9140                                        ;*A WRITE,READ HAS BEEN SUCCESSFULLY COMPLETED
 9141                                        ;*NOW A WRITE CHECK HEADER AND DATA WILL BE GIVEN
 9142                                        ;*FILL THE WRITE CHECK HEADER AND DATA
 9143
 9144  035512                    10$:
 9145
 9146  035512  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9147                                                                  ;R3-RHDS1, R4-RHER1
 9148                                                                  ;GIVE RH-11 INITIALIZE
 9149                                                                  ;SETUP UNIT NUMBER
```

```
9150
9151  035516  004037  043340          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
9152  035522  000005                  5                       ;CYLINDER 5
9153  035524     004          .BYTE   4                       ;SECTOR 4
9154  035525     007          .BYTE   7                       ;TRACK 7
9155  035526  177366                  -262.-4                 ;WORD COUNT (DATA) = 262. +
9156                                                          ;4 HEADER WORDS
9157  035530  002470                  WRFROM                  ;BUS ADDRESS
9158                                                          ;STARTING ADDRESS OF DATA
9159                                                          ;BUFFER = WRFROM
9160  035532  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9161  035534  014000                  ECI!FMT22                       ;16 BITS PER WORD FORMAT
9162                                                          ;INHIBIT ECC CORRECTION
9163                                                          ;DO NOT INHIBIT HEADER COMPARE
9164  035536  002440                  WRCHDT                  ;GET READY TO DO A WRCHDT
9165                                                ;WRITE CHECK HEADER AND DATA WITH 52 IN RHCS1
9166
9167
9168                                  ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9169
9170  035540                  ST25:
9171  035540  004037  041534          JSR     R0,@#SAVER      ;SAVE REGISTERS
9172  035544  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
9173  035546  004512                  SAVERE                  ;STARTING ADDRESS OF WHERE
9174                                                          ;THE REGISTERS ARE SAVED
9175  035550  000022                  18.                     ;NUMBER OF REGISTERS
9176                                                          ;SAVED = 18.
9177
9178  035552  004737  041446          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9179                                                          ;AND THAT NO STATUS BITS ARE STUCK = 1
9180  035556  104400  066402          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
9181                                                          ;THE FIRST SET OF BITS DON'T = 1
9182  035562  000000                  HALT                    ;STOP
9183
9184  035564  013777  004606  144474  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9185                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
9186                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9187                                                          ;'TIME' WILL ONLY SAVE
9188                                                          ;CURRENT CYLINDER ADDRESS
9189                                                          ;AND LOOK AHEAD REGISTERS
9190
9191
9192  035572  013746  002440          MOV     @#WRCHDT,-(SP)  ;GET READY TO MOVE COMMAND
9193  035576  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
9194                                                          ;ENABLE INTERRUPT
9195  035602  012677  144472          MOV     (SP)+,@RHCS1    ;GO WITH
9196                                                          ;52 IN RHCS1 FOR WRITE CHECK HEADER AND DATA
9197                                                          ;WITH INTERRUPT ENABLED
9198  035606  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
9199  035610  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
9200
9201
9202  035612  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
9203  035614  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
9204  035616  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9205  035620  001732                        996.                    ;ALLOW 9860 MICRO SECONDS
9206  035622  001502                        834.                    ;RDY MUST SET BETWEEN
9207                                                                  ;1520 AND 18200 MICRO SECONDS
9208
9209                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
9210                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
9211
9212  035624  013746  002440                MOV     @#WRCHDT,-(SP)  ;SAVE COMMAND
9213  035630  052716  004101                BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
9214  035634  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9215  035640  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
9216                                                                  ;AND COMMAND SHOULD BE SET
9217  035642  001405                        BEQ     64$             ;BRANCH IF GOOD
9218  035644  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
9219  035650  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
9220  035654  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
9221                                                                  ;COMMAND AND IE!DVA!GO SHOULD BE SET
9222  035656  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
9223  035662  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9224  035666  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
9225                                                                  ;SHOULD BE SET
9226  035670  001405                        BEQ     66$             ;BRANCH IF GOOD
9227  035672  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
9228  035676  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
9229  035702  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
9230                                                                  ;MOL!DPR!VV SHOULD BE SET
9231  035704                        66$:
9232
9233                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
9234
9235  035704  004037  041270                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
9236  035710  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
9237  035712  000000                        0                       ;DATA
9238  035714  004037  041270                JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
9239  035720  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
9240  035722  003514                        WRFROM+<266.*2>                 ;DATA
9241  035724  004037  041270                JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
9242  035730  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
9243  035732  003406                        3406                    ;DATA
9244
9245                                        ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
9246                                        ;*WITH REGISTERS AFTER COMMAND
9247
9248
9249  035734  004037  042354                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
9250                                                                  ;PRESENT VALUE
9251  035740  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
9252  035742  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
9253  035744  000022                        18.                     ;18. REGISTERS TO BE COMPARED
9254  035746  035752                        8$                      ;RETURN TO 8$ ON ERROR
9255  035750  035756                        9$                      ;RETURN TO 9$ ON NO ERROR
9256
9257  035752  104040                8$:     ERROR   40              ;WRITE CHECK CAUSED
```

```
9258  035754  000207                          RTS     PC              ;AN IMPROPER REGISTER
9259                                                                  ;CHANGE
9260                                                                  ;GOOD DATA GIVES WHAT
9261                                                                  ;SHOULD BE THERE
9262                                                                  ;RECEIVED DATA GIVES WHAT
9263                                                                  ;WAS THERE AFTER COMMAND
9264
9265
9266  035756                        9$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 46       WRITE CHECK DATA
                                        ;*      THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
                                        ;*      WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
                                        ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
                                        ;*      CONSISTING OF
                                        ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR,
                                        ;*      10 WORDS OF 177777
                                        ;*      10 WORDS OF 0
                                        ;*      10 WORDS OF 052525
                                        ;*      10 WORD OF 125252
                                        ;*      16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
                                        ;*      16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                                        ;*      174 WORDS OF 377
                                        ;*      2 WORDS OF 12345
                                        ;*
                                        ;*
                                        ;*      FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
                                        ;*      THEN THE ABOVE WRITE CHECK DATA IS GIVEN
                                        ;;*****************************************************************
035756  000004                          TST46:  SCOPE
035760  012706  001000                          MOV     #STACK,SP       ;RESET STACK
035764  012737  000046  004604                  MOV     #46,@#TSTNM     ;SAVE TEST NUMBER

035772  004737  041366                          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
035776  004737  041446                          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
036002  104400  066402                          TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                        ;THE FIRST SET OF BITS DON'T = 1
036006  000000                                  HALT                    ;STOP

                                        ;*GET HEADS TO CYLINDER 5


036010  004037  041336                          JSR     R0,@#SEEKCY     ;SEEK FOR
036014  000005                          5                              ;CYLINDER 5

036016  013777  004606  144242                  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


036024  013746  002452                          MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
036030  052716  000101                          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                        ;ENABLE INTERRUPT
036034  012677  144240                          MOV     (SP)+,@RHCS1    ;GO WITH
                                                                         ;4 IN RHCS1 FOR SEEK
```

```
                                                        ;WITH INTERRUPT ENABLED


036040  104412                  WAT                     ;WAIT FOR DRY BIT TO SET
036042  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
036044  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
036046  004704                  2500.                   ;ALLOW 25000 MICRO SECONDS
036050  004704                  2500.                   ;DRY MUST SET BETWEEN
                                                        ;CO AND 50000 MICRO SECONDS


036052  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                        ;*10 WORDS OF EACH 12344,17777,0,52525,125252

036056  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM
036062  002470                  WRFROM                  ;STARTING FROM WRFROM
036064  000012                  10.                     ;10. WORDS
036066  012344                  <5*2000>!<7*40>!4               ;FILL WITH <5*2000>!<7*40>!4

036070  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
036074  002514                  WRFROM+<10.*2>                  ;STARTING FROM WRFROM+<10.*2>
036076  000012                  10.                     ;10. WORDS
036100  177777                  -1                      ;FILL WITH -1

036102  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
036106  002540                  WRFROM+<20.*2>                  ;STARTING FROM WRFROM+<20.*2>
036110  000012                  10.                     ;10. WORDS
036112  000000                  0                       ;FILL WITH 0

036114  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
036120  002564                  WRFROM+<30.*2>                  ;STARTING FROM WRFROM+<30.*2>
036122  000012                  10.                     ;10. WORDS
036124  052525                  52525                   ;FILL WITH 52525

036126  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
036132  002610                  WRFROM+<40.*2>                  ;STARTING FROM WRFROM+<40.*2>
036134  000012                  10.                     ;10. WORDS
036136  125252                  125252                  ;FILL WITH 125252


                        ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>

036140  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
036142  012700  177776          MOV     #177776,R0      ;DATA
036146  012705  000020          MOV     #16.,R5         ;COUNT
036152  012701  002634          MOV     #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
036156  000261                  SEC
036160  010021          1$:     MOV     R0,(R1)+        ;STORE DATA
036162  006100                  ROL     R0              ;GET ZERO ONE BIT LEFT
036164  005305                  DEC     R5              ;COUNT 16
```

```
9375  036166  001374                          BNE     1$              ;BRANCH IF 1$ NOT DONE
9376
9377                                          ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
9378
9379  036170  000241                          CLC
9380  036172  012700  000001                  MOV     #1.R0
9381  036176  010021                  2$:     MOV     R0,(R1)+
9382  036200  006300                          ASL     R0
9383  036202  103375                          BCC     2$
9384  036204  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9385
9386                                          ;*FILL REST OF DATA
9387
9388  036206  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
9389  036212  002734                          WRFROM+<82.*2>                  ;STARTING FROM WRFROM+<82.*2>
9390  036214  000256                          174.                    ;174. WORDS
9391  036216  000377                          377                     ;FILL WITH 377
9392
9393  036220  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
9394  036224  003470                          WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
9395  036226  000002                          2                       ;2 WORDS
9396  036230  012345                          <5*2000>!<7*40>!5                       ;FILL WITH <5*2000>!<7*40>!5
9397
9398
9399                                          ;*FILL THE WRITE CHECK HEADER AND DATA
9400
9401  036232  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
9402  036236  000005                          5                       ;CYLINDER 5
9403  036240     004                  .BYTE   4                       ;SECTOR 4
9404  036241     007                  .BYTE   7                       ;TRACK 7
9405  036242  177400                          -256.                   ;WORD COUNT = 256.
9406  036244  002470                          WRFROM                  ;BUS ADDRESS
9407                                                                  ;STARTING ADDRESS OF DATA
9408                                                                  ;BUFFER = WRFROM
9409  036246  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9410  036250  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
9411                                                                  ;INHIBIT ECC CORRECTION
9412                                                                  ;DO NOT INHIBIT HEADER COMPARE
9413  036252  002436                          WRCHEK                  ;GET READY TO DO A WRCHEK
9414                                                  ;WRITE CHECK DATA WITH 50 IN RHCS1
9415
9416
9417                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9418  036254  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
9419  036260  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
9420  036262  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
9421                                                                  ;THE REGISTERS ARE SAVED
9422  036264  000022                          18.                     ;NUMBER OF REGISTERS
9423                                                                  ;SAVED = 18.
9424
9425  036266  004737  041446                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9426                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
9427  036272  104400  066402                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
9428                                                                  ;THE FIRST SET OF BITS DON'T = 1
```

```
9429  036276  000000                          HALT                    ,STOP
9430
9431  036300  013777  004606  143760          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9432                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
9433                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9434                                                                  ;'TIME' WILL ONLY SAVE
9435                                                                  ;CURRENT CYLINDER ADDRESS
9436                                                                  ;AND LOOK AHEAD REGISTERS
9437
9438
9439  036306  013746  002436                  MOV     @#WRCHEK,-(SP)  ;GET READY TO MOVE COMMAND
9440  036312  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
9441                                                                  ;ENABLE INTERRUPT
9442  036316  012677  143756                  MOV     (SP)+,@RHCS1    ;GO WITH
9443                                                                   ;50 IN RHCS1 FOR WRITE CHECK DATA
9444                                                                  ;WI H INTERRUPT ENABLED
9445  036322  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
9446  036324  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
9447
9448
9449  036326  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
9450  036330  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
9451  036332  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9452  036334  001732                          986.                    ;ALLOW 9860 MICRO SECONDS
9453  036336  001502                          834.                    ;RDY MUST SET BETWEEN
9454                                                                  ;1520 AND 18200 MICRO SECONDS
9455
9456                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
9457                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
9458
9459  036340  013746  002436                  MOV     @#WRCHEK,-(SP)  ;SAVE COMMAND
9460  036344  052716  004101                  BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
9461  036350  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9462  036354  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
9463                                                                  ;AND COMMAND SHOULD BE SET
9464  036356  001405                          BEQ     67$             ;BRANCH IF GOOD
9465  036360  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
9466  036364  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
9467  036370  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
9468                                                                  ;COMMAND AND IE!DVA!GO SHOULD BE SET
9469  036372  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
9470  036376  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9471  036402  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
9472                                                                  ;SHOULD BE SET
9473  036404  001405                          BEQ     69$             ;BRANCH IF GOOD
9474  036406  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
9475  036412  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
9476  036416  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
9477                                                                  ;MOL!DPR!VV SHOULD BE SET
9478  036420                          69$:
9479
9480                                          ,*CHANGE SAVED REGISTERS TO EXPECTED VALUES
9481
9482  036420  004037  041270                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
```

```
 9483  036424  002272                    RHWC                    ;SAVED REGISTER TO CHANGE
 9484  036426  000000                    0                       ;DATA
 9485  036430  004037  041270            JSR     R0,3$#FILLRE    ;MOV WRFROM+<256.*2> INTO SAVED RHBA
 9486  036434  002274                    RHBA                    ;SAVED REGISTER TO CHANGE
 9487  036436  003470                    WRFROM+<256.*2>             ;DATA
 9488  036440  004037  041270            JSR     R0,2$#FILLRE    ;MOV 3405 INTO SAVED RHDST
 9489  036444  002304                    RHDST                   ;SAVED REGISTER TO CHANGE
 9490  036446  003405                    3405                    ;DATA
 9491
 9492                                    ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
 9493                                    ;*WITH REGISTER AFTER COMMAND
 9494
 9495
 9496  036450  004037  042354            JSR     R0,3$#CCMREG    ;COMPARE SAVED REGISTERS WITH
 9497                                                            ;PRESENT VALUE
 9498  036454  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 9499  036[illegible]  002354            WC                      ;TEST DATA STARTING FROM 'RHWC'
 9500  036460  000022                    18.                     ;18. REGISTERS TO BE COMPARED
 9501  036462  036466                    8$                      ;RETURN TO 8$ ON ERROR
 9502  036464  036472                    9$                      ;RETURN TO 9$ ON NO ERROR
 9503  036466  104040            8$:     ERROR   40              ;WRITE CHECK CAUSED
 9504  036470  000207                    RTS     PC              ;AN IMPROPER REGISTER
                                                                ;CHANGE
                                                                ;GOOD DATA GIVES WHAT
                                                                ;SHOULD BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER COMMANDS
 9505
 9506
 9507
 9508
 9509
 9510
 9511
 9512
 9513
 9514
 9515
 9516
 9517
 9518
 9519
 9520
 9521
 9522
 9523
 9524
 9525
 9526
 9527
 9528
 9529
 9530
 9531
 9532
 9533
 9534
 9535
 9536
 9537
 9538
 9539
 9540
 9541
 9542
 9543
 9544
```

```
                                ;;**********************************************************
                                ;*TEST 47       WRITE CHECK DATA USING UNIBUS B
                                ;*      THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
                                ;*      IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
                                ;*      THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
                                ;*      WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
                                ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
                                ;*      CONSISTING OF
                                ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                                ;*      10 WORDS OF 177777
                                ;*      10 WORDS OF 0
                                ;*      10 WORDS OF 052525
                                ;*      10 WORD OF 125252
                                ;*      16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
                                ;*      16 WORDS OF LEFT ROTATING ONE (EG 1.2.4.10)
                                ;*      174 WORDS OF 377
                                ;*      2 WORDS OF 12345
                                ;*
                                ;*
                                ;*      FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
                                ;*      THEN THE ABOVE WRITE CHECK DATA IS GIVEN
                                ;;**********************************************************
036472  000004          TST47:  SCOPE
036474  012706  001000          MOV     #STACK,SP       ;RESET STACK
036500  012737  000047  004634  MOV     #47.,@#TSTNM    ;SAVE TEST NUMBER

036506  004737  041366          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

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

036512  005737  004750          TST     @#RH70          ;TEST FOR RH70 CONTROLLER
036516  001402                  BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
036520  000137  037240          JMP     TST50   ;       JUMP TO NEXT TEST ----------------
036524                  30$:                            ;IF FLAG = 1, DO THIS TEST
036524  004737  041446          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
036530  104400  066432          TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
036534  000000                  HALT                    ;STOP

036536  005737  004732          TST     @#UBUSB         ;IS UNIBUS B THERE
036542  001402                  BEQ     11$             ;UNIBUS B THERE SO CONTINUE
036544  000137  037240          JMP     @#9$            ;NO UNIBUS B, GO TO NEXT TEST

                                ;*GET HEADS TO CYLINDER 5

036550                  11$:

036550  004037  041336          JSR     R0,@#SEEKCY     ;SEEK FOR
036554  000005                  5                       ;CYLINDER 5
```

```
                                ;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
                                ;GO WITH
                                 ;4 IN RHCS1 FOR SEEK
                                ;WITH INTERRUPT ENABLED

                                ;WAIT FOR DRY BIT TO SET
                                ;WAIT FOR RHDS1 REGISTER
                                ;WAIT FOR DRY BIT IN RHDS1 REGISTER
                                ;ALLOW 25000 MICRO SECONDS
                                ;DRY MUST SET BETWEEN
                                ;00 AND 50000 MICRO SECONDS

                                ;SET R1-RHCS1, R2-RHCS2
                                ;R3-RHDS1, R4-RHER1
                                ;GIVE RH-11 INITIALIZE
                                ;SETUP UNIT NUMBER

;*10 WORDS OF EACH 12344,17777,0,52525,125252

                                ;CLEAR 10. WORDS, FROM WRFROM
                                ;STARTING FROM WRFROM
                                ;10. WORDS
                                                ;FILL WITH <5*2000>!<7*40>!4

                                ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
                                                ;STARTING FROM WRFROM+<10.*2>
                                ;10. WORDS
                                ;FILL WITH -1

                                ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
                                                ;STARTING FROM WRFROM+<20.*2>
                                ;10. WORDS
                                ;FILL WITH 0

                                ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
                                                ;STARTING FROM WRFROM+<30.*2>
                                ;10. WORDS
                                ;FILL WITH 52525

                                ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
                                                ;STARTING FROM WRFROM+<40.*2>
```

```
 9622  036674  000012                  10.                     ;10. WORDS
 9623  036676  125252                  125252                  ;FILL WITH 125252
 9624
 9625
 9626                                  ;*FILL LEFT ROTATING ZEROS FROM WRFROM+(50.*2)
 9627
 9628  036700  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9629  036702  012700  177776          MOV     #177776,R0      ;DATA
 9630  036706  012705  000020          MOV     #16.,R5         ;COUNT
 9631  036712  012701  002634          MOV     #WRFROM+(50.*2),R1 ;WHERE DATA GOES
 9632  036716  000261                  SEC
 9633  036720  010021          1$:     MOV     R0,(R1)+        ;STORE DATA
 9634  036722  006100                  ROL     R0              ;GET ZERO ONE BIT LEFT
 9635  036724  005305                  DEC     R5              ;COUNT 16
 9636  036726  001374                  BNE     1$              ;BRANCH IF 16 NOT DONE
 9637
 9638                                  ;*FILL LEFT ROTATING ONE INTO WRFROM+(65.*2)
 9639
 9640  036730  000241                  CLC
 9641  036732  012700  000001          MOV     #1,R0
 9642  036736  010021          2$:     MOV     R0,(R1)+
 9643  036740  006300                  ASL     R0
 9644  036742  103375                  BCC     2$
 9645  036744  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 9646
 9647                                  ;*FILL REST OF DATA
 9648  036746  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+(82.*2)
 9649  036752  002734                  WRFROM+(82.*2)                  ;STARTING FROM WRFROM+(82.*2)
 9650  036754  000256                  174.                    ;174. WORDS
 9651  036756  000377                  377                     ;FILL WITH 377
 9652
 9653  036760  004037  041236          JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+(256.*2)
 9654  036764  003470                  WRFROM+(256.*2)                 ;STARTING FROM WRFROM+(256.*2)
 9655  036766  000002                  2                       ;2 WORDS
 9656  036770  012345                  (5*2000)!(7*40)!5               ;FILL WITH (5*2000)!(7*40)!5
 9657
 9658
 9659                                  ;*FILL THE WRITE CHECK HEADER AND DATA
 9660
 9661  036772  004037  043340          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9662  036776  000005                  5                       ;CYLINDER 5
 9663  037000      004          .BYTE   4                       ;SECTOR 4
 9664  037001      007          .BYTE   7                       ;TRACK 7
 9665  037002  177400                  -256.                   ;WORD COUNT = 256.
 9666  037004  002470                  WRFROM                  ;BUS ADDRESS
 9667                                                          ;STARTING ADDRESS OF DATA
 9668                                                          ;BUFFER = WRFROM
 9669  037006  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9670  037010  014000                  ECI!FMT22                       ;16 BITS PER WORD FORMAT
 9671                                                          ;INHIBIT ECC CORRECTION
 9672                                                          ;DO NOT INHIBIT HEADER COMPARE
 9673  037012  002436                  WRCHEK                  ;GET READY TO DO A WRCHEK
 9674                                                  ;WRITE CHECK DATA WITH SC IN RHCS1
 9675
```

```
                                ;*SET PORT B
                                ;THAT IS UNIBUS B

                                ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
037022  004037  041534          JSR     R0,@#SAVER      ;SAVE REGISTERS
037026  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
037030  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
                                                        ;THE REGISTERS ARE SAVED
037032  000022                  18.                     ;NUMBER OF REGISTERS
                                                        ;SAVED = 18.

037034  004737  041446          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
037040  104400  066402          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
037044  000000                  HALT                    ;STOP

037046  013777  004606  143212  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

                                ;*SET PORT SELECT

037054  013746  002436          MOV     @#WRCHEK,-(SP)  ;GET READY TO MOVE COMMAND
037060  052716  002101          BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
037064  012677  143210          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;SO IN RHCS1 FOR WRITE CHECK DATA
                                                        ;WITH INTERRUPT ENABLED
037070  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
037072  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION


037074  104412                  WAT                     ;WAIT FOR RDY BIT TO SET
037076  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
037100  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
037102  001732                  986.                    ;ALLOW 9860 MICRO SECONDS
037104  001502                  934.                    ;RDY MUST SET BETWEEN
                                                        ;1520 AND 18200 MICRO SECONDS

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

037106  013746  002436          MOV     @#WRCHEK,-(SP)  ;SAVE COMMAND
037112  052716  006101          BIS     #IE!DVA!PSEL!GO,(SP)    ;INCLUDE IE!DVA!PSEL!GO
037116  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
037122  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!PSEL!GO
                                                        ;AND COMMAND SHOULD BE SET
037124  001405                  BEQ     67$             ;BRANCH IF GOOD
```

```
9730  037126  310037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
9731  037132  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
9732  037136  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
9733                                                          ;COMMAND AND IE!DVA!PSEL!GO SHOULD BE SET
9734  037140  012746  010500    57$:  MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
9735  037144  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9736  037150  022605                  CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
9737                                                          ;SHOULD BE SET
9738  037152  001405                  BEQ     69$             ;BRANCH IF GOOD
9739  037154  010537  001126          MOV     R5,@#$BDDAT     ;BAD DATA
9740  037160  010337  004600          MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
9741  037164  104063                  ERROR   63              ;DURING ABOVE OPERATION ONLY
9742                                                          ;MOL!DPR!VV SHOULD BE SET
9743  037166                    69$:
9744
9745                                  ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
9746
9747  037166  004037  041270          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
9748  037172  002272                  RHWC                    ;SAVED REGISTER TO CHANGE
9749  037174  000000                  0                       ;DATA
9750  037176  004037  041270          JSR     R0,@#FILLRE     ;MOV WRFROM+(256.*2) INTO SAVED RHBA
9751  037202  002274                  RHBA                    ;SAVED REGISTER TO CHANGE
9752  037204  003470                  WRFROM+(256.*2)                 ;DATA
9753  037206  004037  041270          JSR     R0,@#FILLRE     ;MOV 3405 INTO SAVED RHDST
9754  037212  002304                  RHDST                   ;SAVED REGISTER TO CHANGE
9755  037214  003405                  3405                    ;DATA
9756
9757                                  ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
9758                                  ;*WITH REGISTER AFTER COMMAND
9759
9760  037216  004037  042354          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
9761                                                          ;PRESENT VALUE
9762  037222  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
9763  037224  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
9764  037226  000022                  18.                     ;18. REGISTERS TO BE COMPARED
9765  037230  037234                  8$                      ;RETURN TO 8$ ON ERROR
9766  037232  037240                  9$                      ;RETURN TO 9$ ON NO ERROR
9767
9768  037234  104076            8$:   ERROR   76              ;WHILE USING UNIBUS B
9769                                                          ;WRITE CHECK CAUSED
9770  037236  000207                  RTS     PC              ;AN IMPROPER REGISTER
9771                                                          ;CHANGE
9772                                                          ;GOOD DATA GIVES WHAT
9773                                                          ;SHOULD BE THERE
9774                                                          ;RECEIVED DATA GIVES WHAT
9775                                                          ;WAS THERE AFTER COMMANDS
9776
9777  037240                    9$:
9778
9779
9780
9781
9782
```

```
                                ;;*****************************************************************
                                ;*TEST 50      WRITE PROTECT OPERATION
                                ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
                                ;*
                                ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                ;*      THEN THIS TEST IS NOT PERFORMED
                                ;*
                                ;*      IF NO ACT-11 MONITOR IS PRESENT
                                ;*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
                                ;*
                                ;*      WRITE FROM BUFFER IS FILLED WITH ALL ONES AND
                                ;*      SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH
                                ;*      ALL ONES
                                ;*      ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS
                                ;*      PRESSED AND ALL REGISTERS ARE CHECKED.
                                ;*      WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS
                                ;*      ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0  70. WORDS
                                ;*      ALL REGISTERS ARE CHECKED
                                ;*      THE SAME SECTOR IS READ AND DATA COMPARED TO SEE
                                ;*      THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL
                                ;*      ONES AND NOT 377)
                                ;*      THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK
                                ;*      WRITE LOCKS AND ALL REGISTERS ARE COMPARED
                                ;;*****************************************************************
037240  000004                TST50:  SCOPE


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

037242  005737  004724                TST     @#NOPUSH        ;IS THIS A 220 START ?
037246  001007                        BNE     1$              ;SKIP THIS TEST IF SO
037250  005737  000042                TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
037254  001004                        BNE     1$              ;SKIP THIS TEST
037256  005737  001100                TST     @#$PASS         ;FIRST PASS ?
037262  001001                        BNE     1$              ;SKIP THIS TEST IF NOT
037264  000402                        BR      2$              ;CONTINUE WITH THIS TEST

037266                          1$:
037266  000137  040464                JMP     TST51   ;       JUMP TO NEXT TEST ----------
037272                          2$:
037272  012706  001000                MOV     #STACK,SP       ;RESET STACK
037276  012737  000050  004604        MOV     #50.,@#TSTNM    ;SAVE TEST NUMBER

037304  004737  041366                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER

                                        ;*FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES
                                        ;*FILL WRITE FROM BUFFER
037310  004037  041236                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
```

```
 9837  037314  002470                          WRFROM                  ;STARTING FROM WRFROM
 9838  037316  000400                          256.                    ;256. WORDS
 9839  037320  177777                          -1                      ;FILL WITH -1
 9840
 9841
 9842                                          ;*FILL WRITE DATA COMMAND
 9843
 9844  037322  004037  043340                  JSR     RD.3*RUN        ;SETUP TO RUN FOR DATA COMMAND
 9845  037326  000000                          0                       ;CYLINDER 0
 9846  037330     000                .BYTE     0                       ;SECTOR 0
 9847  037331     000                .BYTE     0                       ;TRACK 0
 9848  037332  177400                          -256.                   ;WORD COUNT = 256.
 9849  037334  002470                          WRFROM                  ;BUS ADDRESS
 9850                                                                  ;STARTING ADDRESS OF DATA
 9851                                                                  ;BUFFER = WRFROM
 9852  037336  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9853  037340  010000                          FMT22                   ;16 BITS PER WORD FORMAT
 9854                                                                  ;DO NOT INHIBIT ECC CORRECTION
 9855                                                                  ;DO NOT INHIBIT HEADER COMPARE
 9856  037342  002442                          WRIDAT                  ;GET READY TO DO A WRIDAT
 9857                                                  ;WRITE DATA WITH 60 IN RHCS1
 9858
 9859
 9860  037344  004737  041446                  JSR     PC,3*CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 9861                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
 9862  037350  104400  066402                  TYPE    .CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 9863                                                                  ;THE FIRST SET OF BITS DON'T = 1
 9864  037354  000000                          HALT                    ;STOP
 9865
 9866  037356  013777  004606  142702          MOV     3*RP4VEC,3RPVEC ;SET RP04 VECTOR ADDRESS
 9867                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9868                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 9869                                                                  ;'TIME' WILL ONLY SAVE
 9870                                                                  ;CURRENT CYLINDER ADDRESS
 9871                                                                  ;AND LOOK AHEAD REGISTERS
 9872
 9873
 9874  037364  013746  002442                  MOV     3*WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
 9875  037370  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 9876                                                                  ;ENABLE INTERRUPT
 9877  037374  012677  142700                  MOV     (SP)+,3RHCS1    ;GO WITH
 9878                                                                   ;60 IN RHCS1 FOR WRITE DATA
 9879                                                                  ;WITH INTERRUPT ENABLED
 9880
 9881                                          ;*TIME IS NOT CRITICAL
 9882
 9883  037400  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
 9884  037402  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9885  037404  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9886  037406  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
 9887  037410  004704                          2500.                   ;RDY MUST SET BETWEEN
 9888                                                                  ;00 AND 50000 MICRO SECONDS
 9889
 9890                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT
```

```
                                        ;*BUTTON HAS BEEN HIT
                                        ;
037412  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
037416  002372                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
037420  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
037422  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.
                                        ;
037424  104400  037432                  TYPE    ,68$            ;;TYPE ASCIZ STRING
037430  000407                          BR      67$             ;;GET OVER THE ASCIZ
                                ;;68$:   .ASCIZ  <15><12>/ON DRIVE /
037450                          67$:
037450  013746  004716                  MOV     @#UNIT,-(SP)            ;GET UNIT UNDER TEST
037454  104404                          TYPDS
037456  104400  037464                  TYPE    ,70$            ;;TYPE ASCIZ STRING
037462  000442                          BR      69$             ;;GET OVER THE ASCIZ
                                ;;70$:   .ASCIZ  <15><12>/PUSH WRITE PROTECT BUTTON TO LOCK OUT WRITES THEN HIT CONTINUE/
037570                          69$:
037570  000000                          HALT
                                        ;*THE ONLY REGISTER THAT SHOULD  CHANGE IS RHDS1 - BIT #11
                                        ;*-WRL
                                        ;
037572  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
037576  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
                                        ;
037600  000001                          1                       ;1 BIT/BITS TO BE CHANGED
037602  000001                          1                       ;NEW VALUE OF WRL IS 1
037604  004000                          WRL                     ;CHANGE WRL BIT
                                        ;
                                        ;*COMPARE ALL REGISTERS BEFORE WRITE WAS LOCKED
                                        ;*OUT WITH REGISTER VALUES AFTER WRITE WAS LOCKED OUT
                                        ;
037606  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
037612  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
037614  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
037616  000022                          18.                     ;18. REGISTERS TO BE COMPARED
037620  037624                          3$                      ;RETURN TO 3$ ON ERROR
037622  037630                          4$                      ;RETURN TO 4$ ON NO ERROR
                                        ;
037624  104041                  3$:     ERROR   41              ;LOCKING OUT WRITE BY
037626  000207                          RTS     PC              ;WRITE LOCK BUTTON CAUSED
                                                                ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER WRITE
                                                                ;WAS LOCKED OUT BY BUTTON
                                        ;
                                        ;*NOW A WRITE WILL BE ATTEMPTED WITH WRITE LOCKED
                                        ;*OUT BY BUTTON
                                        ;*FILL WRITE FROM BUFFER WITH 377
                                        ;
```

```
 9945  037630  013737  037630  001110  4$:     MOV     @#4$,@#$LPERR   ;SCOPE LOOP STARTS FROM HERE
 9946
 9947
 9948  037636  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9949                                                                  ;R3-RHDS1, R4-RHER1
 9950                                                                  ;GIVE RH-11 INITIALIZE
 9951                                                                  ;SETUP UNIT NUMBER
 9952  037642  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
 9953  037646  002470                          WRFROM                  ;STARTING FROM WRFROM
 9954  037650  000400                          256.                    ;256. WORDS
 9955  037652  000377                          377                     ;FILL WITH 377
 9956
 9957
 9958                                          ;*TRY A ONE WORD WRITE
 9959
 9960  037654  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9961  037660  000000                          0                       ;CYLINDER 0
 9962  037662     000                  .BYTE   0                       ;SECTOR 0
 9963  037663     000                  .BYTE   0                       ;TRACK 0
 9964  037664  177777                          -1.                     ;WORD COUNT = 1.
 9965  037666  002470                          WRFROM                  ;BUS ADDRESS
 9966                                                                  ;STARTING ADDRESS OF DATA
 9967                                                                  ;BUFFER = WRFROM
 9968  037670  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9969  037672  010000                          FMT22                   ;16 BITS PER WORD FORMAT
 9970                                                                  ;DO NOT INHIBIT ECC CORRECTION
 9971                                                                  ;DO NOT INHIBIT HEADER COMPARE
 9972  037674  002442                          WRIDAT                  ;GET READY TO DO A WRIDAT
 9973                                                  ;WRITE DATA WITH 60 IN RHCS1
 9974
 9975
 9976                                          ;*SAVE REGISTERS
 9977  037676  004037  041534                  JSR     R0,@#SAVER      ;SAVE REGISTERS
 9978  037702  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 9979  037704  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
 9980                                                                  ;THE REGISTERS ARE SAVED
 9981  037706  000022                          18.                     ;NUMBER OF REGISTERS
 9982                                                                  ;SAVED = 18.
 9983
 9984  037710  013777  004606  142350          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 9985                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9986                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 9987                                                                  ;'TIME' WILL ONLY SAVE
 9988                                                                  ;CURRENT CYLINDER ADDRESS
 9989                                                                  ;AND LOOK AHEAD REGISTERS
 9990
 9991
 9992  037716  013746  002442                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
 9993  037722  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 9994                                                                  ;ENABLE INTERRUPT
 9995  037726  012677  142346                  MOV     (SP)+,@RHCS1    ;GO WITH
 9996                                                                   ;60 IN RHCS1 FOR WRITE DATA
 9997                                                                  ;WITH INTERRUPT ENABLED
 9998                                          ;*TIME IS NOT CRITICAL
```

```
 9999
10000  037732  104412                          WAT                     ;WAIT FOR RDY BIT TO SET
10001  037734  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
10002  037736  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10003  037740  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
10004  037742  004704                          2500.                   ;RDY MUST SET BETWEEN
10005                                                                  ;00 AND 50000 MICRO SECONDS
10006
10007                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
10008
10009  037744  017737  142322  004612          MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
10010  037752  017737  142316  004614          MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
10011  037760  017746  142312                  MOV     @RHCS2,-(SP)    ;GET RHCS2
10012  037764  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
10013  037770  042737  000300  004616          BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
10014  037776  052637  004616                  BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
10015
10016  040002  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10017  040006  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
10018
10019  040010  000002                          2                       ;2 BIT/BITS TO BE CHANGED
10020  040012  000001                          1                       ;NEW VALUE OF SC IS 1
10021  040014  100000                          SC                      ;CHANGE SC BIT
10022  040016  000001                          1                       ;NEW VALUE OF TRE IS 1
10023  040020  040000                          TRE                     ;CHANGE TRE BIT
10024  040022  004037  041270                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
10025  040026  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
10026  040030  000001                          1                       ;DATA
10027
10028  040032  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10029  040036  002302                          RHER1                   ;CHANGE RHER1 REGISTER
10030
10031  040040  000001                          1                       ;1 BIT/BITS TO BE CHANGED
10032  040042  000001                          1                       ;NEW VALUE OF WLE IS 1
10033  040044  004000                          WLE                     ;CHANGE WLE BIT
10034
10035  040046  004037  042246                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10036  040052  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
10037
10038  040054  000002                          2                       ;2 BIT/BITS TO BE CHANGED
10039  040056  000001                          1                       ;NEW VALUE OF ATA IS 1
10040  040060  100000                          ATA                     ;CHANGE ATA BIT
10041  040062  000001                          1                       ;NEW VALUE OF ERR IS 1
10042  040064  040000                          ERR                     ;CHANGE ERR BIT
10043
10044  040066  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
10045                                                                  ;FOR WORKING DRIVE IN
10046                                                                  ;SAVED RHAS LOACTION
10047
10048                                          ;*COMPARE REGISTERS BEFORE WRITE WAS ATTEMPTED
10049                                          ;*WITH REGISTERS AFTER ATTEMPT
10050
10051
10052  040074  004037  042354                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
10053
10054  040100  004612                          SAVERE                  ;PRESENT VALUE
10055  040102  002354                          WC                      ;GOOD DATA SAVED IN 'SAVERE'
10056  040104  000022                          18.                     ;TEST DATA STARTING FROM 'RHWC'
10057  040106  040112                          5$                      ;18. REGISTERS TO BE COMPARED
10058  040110  040116                          6$                      ;RETURN TO 5$ ON ERROR
10059                                                                  ;RETURN TO 6$ ON NO ERROR
10060  040112  104042                  5$:     ERROR   42              ;ATTEMPTING TO WRITE WITH
10061  040114  000207                          RTS     PC              ;WRITE LOCKED OUT
10062                                                                  ;CAUSED IMPROPER REGISTER
10063                                                                  ;CHANGE
10064                                                                  ;GOOD DATA GIVES WHAT SHOULD
10065                                                                  ;BE THERE
10066                                                                  ;RECEIVED DATA GIVES WHAT WAS
10067                                                                  ;THERE AFTER ATTEMPT
10068
10069                                          ;*NOW A READ WILL BE DONE TO DETERMIN THAT
10070                                          ;*READS CAN BE DONE WITH WRITE LOCKED OUT AND
10071                                          ;*THAT NO DATA ON DISK GOT CHANGED, BUT FIRST CLEAR ERROR
10072
10073  040116                          6$:
10074
10075  040116  004737  041366                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10076                                                                  ;R3-RHDS1, R4-RHER1
10077                                                                  ;GIVE RH-11 INITIALIZE
10078                                                                  ;SETUP UNIT NUMBER
10079
10080                                          ;*FILL READ INTO BUFFER WITH 0
10081  040122  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
10082  040126  003534                          REINTO                  ;STARTING FROM REINTO
10083  040130  000400                          256.                    ;256. WORDS
10084  040132  000000                          0                       ;FILL WITH 0
10085
10086
10087                                          ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
10088  040134  004037  041236                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
10089  040140  002470                          WRFROM                  ;STARTING FROM WRFROM
10090  040142  000400                          256.                    ;256. WORDS
10091  040144  177777                          -1                      ;FILL WITH -1
10092
10093
10094                                          ;*FILL COMMAND
10095
10096  040146  004037  043340                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
10097  040152  000000                          0                       ;CYLINDER 0
10098  040154     000                  .BYTE   0                       ;SECTOR 0
10099  040155     000                  .BYTE   0                       ;TRACK 0
10100  040156  177400                          -256.                   ;WORD COUNT = 256.
10101  040160  003534                          REINTO                  ;BUS ADDRESS
10102                                                                  ;STARTING ADDRESS OF DATA
10103                                                                  ;BUFFER = REINTO
10104  040162  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10105  040164  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
10106                                                                  ;INHIBIT ECC CORRECTION
```

```
10107
10108  040166  002446                          READAT                  ;DO NOT INHIBIT HEADER COMPARE
10109                                                                   ;GET READY TO DO A READAT
10110                                                          ;READ DATA WITH 70 IN RHCS1
10111
10112  040170  013777  004606  142070           MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10113                                                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
10114                                                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10115                                                                   ;'TIME' WILL ONLY SAVE
10116                                                                   ;CURRENT CYLINDER ADDRESS
10117                                                                   ;AND LOOK AHEAD REGISTERS
10118
10119
10120  040176  013746  002446                   MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
10121  040202  052716  000101                   BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
10122                                                                   ;ENABLE INTERRUPT
10123  040206  012677  142066                   MOV     (SP)+,@RHCS1    ;GO WITH
10124                                                                    ;70 IN RHCS1 FOR READ DATA
10125                                                                   ;WITH INTERRUPT ENABLED
10126
10127
10128  040212  104412                           WAT                     ;WAIT FOR RDY BIT TO SET
10129  040214  002300                           RHCS1                   ;WAIT FOR RHCS1 REGISTER
10130  040216  000200                           RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10131  040220  001614                           908.                    ;ALLOW 9080 MICRO SECONDS
10132  040222  001507                           839.                    ;RDY MUST SET BETWEEN
10133                                                                   ;690 AND 17470 MICRO SECONDS
10134
10135                                          ;*COMPARE READ DATA
10136
10137  040224  004037  043404                   JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
10138  040230  002470                           WRFROM                  ;GOOD DATA STARTS FROM WRFROM
10139  040232  003534                           REINTO                  ;TEST DATA STARTS FROM REINTO
10140  040234  000400                           256.                    ;256. WORDS TO BE COMPARED
10141  040236  040242                           7$                      ;RETURN TO 7$ ON ERROR
10142  040240  040244                           8$                      ;RETURN TO 8$ ON NO ERROR
10143
10144
10145  040242  104043                  7$:      ERROR   43              ;WRITING WITH WRITE
10146                                                                   ;LOCKED CHANGED DISK
10147                                                                   ;GOOD DATA GIVES WHAT WAS
10148                                                                   ;ON DISK BEFORE WRITE WITH
10149                                                                   ;WRITE LOCK WAS ATTEPTED
10150                                                                   ;
10151                                                                   ;RECEIVED DATA GIVES WHAT
10152                                                                   ;WAS READ BACK AFTER WRITE
10153                                                                   ;WITH WRITE LOCKED WAS ATTEPTED
10154
10155                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE LOCK WAS BEEN
10156                                          ;*UNLOCKED
10157
10158  040244                          8$:
10159  040244  004037  041534                   JSR     R0,@#SAVER      ;SAVE REGISTERS
10160  040250  002272                           RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
```

```
                                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
                                        18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.


                                ST20:
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      ,64$            ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/ON DRIVE /
                                ,64$:
                                        MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
                                        TYPDS
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      ,66$            ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/PUSH WRITE PROTCT BUTTON TO UNLOCK WRITES THEN HIT CONTINUE/<15
                                ,66$:
                                        HALT

                                        ;*THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1

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

                                        1                       ;1 BIT/BITS TO BE CHANGED
                                        0                       ;NEW VALUE OF WRL IS 0
                                        WRL                     ;CHANGE WRL BIT

                                        ;*COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
                                        ;*WITH REGISTERS AFTER WRITE WAS UNLOCKED


                                        JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
                                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
                                        WC                      ;TEST DATA STARTING FROM 'RHWC'
                                        19.                     ;!8. REGISTERS TO BE COMPARED
                                        9$                      ;RETURN TO 9$ ON ERROR
                                        10$                     ;RETURN TO 10$ ON NO ERROR

                                9$:     ERROR   44              ;UNLOCKING WRITES BY WRITE
                                        RTS     PC              ;LOCK BUTTON CAUSED AN ERROR
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER WRITES WERE
                                                                ;UNLOCKED
                                                                ;ON THIS ERROR NO LOOPING IS RECOMMENDED
                                                                ;JUST A HALT ON ERROR WILL DO THE SAME
                                                                ;THING AS ONLY THE REGISTERS ARE READ
                                10$:    MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
```

```
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 51        END OF DRIVE

                                        ;*       THIS IS THE END OF TEST FOR ONE DRIVE

                                        ;*       IF THERE ARE MORE DRIVES, THEN THE PROGRAM
                                        ;*       JUMPS TO TEST 4 FOR TESTING THE NEXT DRIVE

                                        ;*       END PASS IS REACHED ONLY AFTER ALL DRIVES ARE TESTED

                                        ;;*****************************************************************
010236  040464  000004                  TST51:  SCOPE
010237  040466  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
010238  040474  012737  000000  177776          MOV     #0,PS           ;REINSTATE PS TO 0
010239
010240  040502  104400  040510                  TYPE    ,65$            ;;TYPE ASCIZ STRING
010241  040506  000424                          BR      64$             ;;GET OVER THE ASCIZ
010242                                  ;;65$:  .ASCIZ  <15><12>/TOTAL ERRORS ON THIS PASS ON UNIT NO./
010243  040560                          64$:
010244  040560  013746  004716                  MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
010245  040564  104404                          TYPOS
010246  040566  104400  040574                  TYPE    ,67$            ;;TYPE ASCIZ STRING
010247  040572  000402                          BR      66$             ;;GET OVER THE ASCIZ
010248                                  ;;67$:  .ASCIZ  / =/
010249  040600                          66$:
010250  040600  013746  001112                  MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
010251  040604  104404                          TYPOS
010252  040606  005037  001112                  CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
010253  040612  005037  001102                  CLR     @#$TSTNM        ;CLEAR TEST NUMBER
010254  040616  005737  004726                  TST     @#SELECT        ;STARTING FROM 200 ?
010255  040622  001413                          BEQ     3$              ;CHECK NEXT DRIVE IF SO
010256                                                                  ;CONTINUE WITH THIS ONE IF NOT
010257
010258  040624  005237  001100                  INC     @#$PASS         ;INCREASE PASS COUNT
010259  040630  104400  041021                  TYPE    ,$ENDMG         ;TYPE "END PASS #"
010260  040634  013746  001100                  MOV     @#$PASS,-(SP)
010261  040640  104404                          TYPOS
010262  040642  104400  041016                  TYPE    ,$ENULL
010263  040646  000137  007744                  JMP     @#TST4          ;JUMP TEST 4 ----------------------
010264
010265  040652  012737  177777  046760  3$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
010266  040660  005337  004720                  DEC     @#NOUNITS       ;NO. OF UNITS PRESENT
010267  040664  001413                          BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
010268  040666  013700  004716                  MOV     @#UNIT,R0       ;UNIT UNDER TEST
010269  040672  012701  004676                  MOV     #UNITS,R1       ;TABLE POINTER
010270  040676  022100                  1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED ?
010271  040700  001401                          BEQ     2$              ;BRANCH IF YES
010272  040702  000775                          BR      1$              ;BRANCH IF NO
010273  040704  011137  004716          2$:     MOV     (R1),@#UNIT     ;MAKE THIS NEXT UNIT
010274  040710  000137  007744                  JMP     @#TST4          ;TEST THE NEXT DRIVE --------------
```

```
:0265
```

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


                                .SBTTL  END OF PASS ROUTINE

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

040714                          $EOP:
040714  000004                         SCOPE
040716  005037  001102                 CLR     $TSTNM          ;;ZERO THE TEST NUMBER
040722  005037  001212                 CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
040726  005237  001100                 INC     $PASS           ;;INCREMENT THE PASS NUMBER
040732  042737  100000  001100         BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
040740  005327                         DEC     (PC)+           ;;LOOP?
040742  000001                 $EOPCT: .WORD   1
040744  003022                         BGT     $DOAGN          ;;YES
040746  012737                         MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
040750  000001                 $ENDCT: .WORD   1
040752  040742                         $EOPCT
040754  104401  041021                 TYPE    ,$ENDMG         ;;TYPE "END PASS #"
040760  013746  001100                 MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
040764  104404                         TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
040766  104401  041016                 TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
040772  013700  000042         $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
040776  001405                         BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
041000  000005                         RESET                   ;;CLEAR THE WORLD
041002  004710                 $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
041004  000240                         NOP                     ;;SAVE ROOM
041006  000240                         NOP                     ;;FOR
041010  000240                         NOP                     ;;ACT11
041012                          $DOAGN:
041012  000137                         JMP     @(PC)+          ;;RETURN
041014  006312                 $RTNAD: .WORD   TST1
041016      377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
041021      015  042412  042116 $ENDMG: .ASCIZ  <15><12>"END PASS #"
041026  050040  051501  020123
041034  000043
```

```
                                        .SBTTL  JAM CURRENT CYLINDER ROUTINE

                                ;*THIS ROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER - 'RHCC'
                                ;*BY GIVING A 'SEEK' COMMAND FOLLOWED BY AN INIT WHICH WILL LOAD
                                ;*'RHCC' WITH THE DESIRED CYLINDER VALUE.  THE ROUTINE THEN CHECKS
                                ;*THAT THE LOADED VALUE IS CORRECT.
                                ;*
                                ;*CALL IS:
                                ;*      JSR     R0,@#MAKECYL
                                ;*      XC                              ;DESIRED VALUE OF CURRENT CYLINDER
                                
041036                          MAKECYL:
041036  010546                         MOV     R5,-(SP)                ;;PUSH R5 ON STACK
041040  010037  041422                 MOV     R0,@#PCJSR              ;PC OF JSR+4
041044  162737  000004  041422         SUB     #4,@#PCJSR              ;SAVE PC OF JSR
041052  012005                         MOV     (R0)+,R5                ;GETTING READY TO FILL DESIRED CYLINDER
041054  010577  141232                 MOV     R5,@RHCA                ;FILL DESIRED CYLINDER REGISTER
041060  005077  141220                 CLR     @RHDST                  ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
041064  013777  002452  141206         MOV     @#SEECOM,@RHCS1         ;FILL SEEK COMMAND
041072  012777  000001  141220         MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE

041100  004737  043300                 JSR     PC,@#PUTREG             ;TAKE A REGISTER SNAPSHOT
041104  032737  040000  002404         BIT     #ERR,@#DS1              ;CHECK FOR COMPOSITE ERROR
041112  001401                         BEQ     2$                      ;NOT = 1, A-OK
041114  104103                         ERROR   103                     ;REGISTER CONTENTS INCORRECT BEFORE A
                                                                        ;DIAGNOSTIC SEEK

041116  052777  000001  141154 2$:     BIS     #GO,@RHCS1              ;ISSUE 'GO' TO SEEK COMMAND
041124  000240                         NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
041126  000240                         NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
041130  000240                         NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
041132  000240                         NOP                             ;ALLOW TIME FOR SEEK TO HANG UP

041134  004737  043300                 JSR     PC,@#PUTREG             ;TAKE A 2ND REGISTER SNAPSHOT
041140  032737  040000  002404         BIT     #ERR,@#DS1              ;CHECK FOR ERRORS
041146  001401                         BEQ     3$                      ;NOT = 1, A-OK
041150  104104                         ERROR   104                     ;REGISTER CONTENTS INCORRECT AFTER
                                                                        ;A DIAGNOSTIC SEEK

041152  004737  041366         3$:     JSR     PC,@#CLDISK             ;GIVE INIT TO FORCE THE TRANSFER
041156  017737  141152  001126         MOV     @RHCC,@#$BDDAT          ;TEST DATA
041164  020537  001126                 CMP     R5,@#$BDDAT             ;COMPARE CURRENT CYLINDER
041170  001406                         BEQ     1$                      ;BRANCH IF GOOD
041172  010537  001124                 MOV     R5,@#$GDDAT             ;GOOD VALUE OF RHCC
041176  013737  002334  004600         MOV     @#RHCC,@#REGADR         ;FAILING REGISTER ADDRESS
041204  104077                         ERROR   77                      ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
                                                                        ;REGISTER AFTER A SEEK AND AN INIT

041206                          1$:
041206  012605                         MOV     (SP)+,R5                ;;POP STACK INTO R5
041210  000300                         RTS     R0
```

```
10362
10363                                   ;*THIS FILLS MEMORY WITH GIVEN DATA
10364                                   ;*USED CHIEFLY FOR HEADER INFORMATION
10365                                   ;*CALL IS
10366                                   ;*      JSR     R0,@#FLHEAD     ;FILL HEADER
10367                                   ;*      LOC                     ;LOCATION WHERE SAVED
10368                                   ;*      XN                      ;NUMBER OF WORDS
10369                                   ;*      XD1                     ;DATA REPEATED XN TIMES
10370                                   ;*      XD2                     ;DATA REPEATED XN TIMES
10371                                   ;*
10372                                   ;*
10373                                   ;*
10374
10375  041212                           FLHEAD:
10376  041212  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10377  041214  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10378  041216  012001                           MOV     (R0)+,R1        ;R1 HAS ADDRESS OF WHERE TO SAVE
10379  041220  012002                           MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS
10380
10381                                           ;*NOW FILL DATA
10382
10383  041222  012021                   1$:     MOV     (R0)+,(R1)+     ;SAVE DATA
10384  041224  005302                           DEC     R2              ;DECREMENT COUNT
10385  041226  001375                           BNE     1$              ;BRANCH IF INCOMPLETE
10386  041230  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
10387  041232  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
10388  041234  000200                           RTS     R0
10389
10390
10391
10392                                   ;*THIS CLEARS ANY BLOCK OF MEMORY.
10393                                   ;*FILLING IT WITH ANY DATA
10394                                   ;*CALL IS
10395                                   ;*      JSR     R0,@#CLAREA
10396                                   ;*      F                       ;FROM
10397                                   ;*      N                       ;NUMBER OF WORDS
10398                                   ;*      D                       ;DATA  TO BE FILLED
10399                                   ;*
10400                                   ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
10401                                   ;*R2 WILL HAVE NUMBER OF WORDS
10402                                   ;*R3 WILL HAVE DATA
10403
10404  041236                           CLAREA:
10405  041236  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10406  041240  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10407  041242  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10408  041244  012001                           MOV     (R0)+,R1        ;FROM
10409  041246  012002                           MOV     (R0)+,R2        ;NUMBER
10410  041250  012003                           MOV     (R0)+,R3        ;DATA
10411  041252  010321                   1$:     MOV     R3,(R1)+        ;MOVE DATA
10412  041254  005302                           DEC     R2              ;COUNT
10413  041256  001375                           BNE     1$              ;BRANCH IF NOT COMPLETE
10414  041260  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
10415  041262  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 10416  041264  012603:                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 10417  041266  000200                         RTS     R0              ;RETURN TO MAIN PROGRAM
 10418
 10419
 10420
 10421
 10422
 10423
 10424
 10425
 10426
 10427
 10428
 10429
 10430
 10431                                  ;*THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
 10432                                  ;*WITH GIVEN VALUE
 10433                                  ;*CALL IS
 10434                                  ;*      JSR     R0,3$#FILLRE
 10435                                  ;*      RHXX            ;REGISTER NAME
 10436                                  ;*      D               ;DATA
 10437                                  ;*
 10438  041270                          FILLRE:
 10439  041270  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 10440  041272  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 10441  041274  012001                         MOV     (R0)+,R1        ;ADDRESS OF ADDRESS OF REGISTER
 10442  041276  012002                         MOV     (R0)+,R2        ;DATA
 10443  041300  162701  002272                 SUB     #RHWC,R1        ;OFSET
 10444  041304  010261  004612                 MOV     R2,SAVERE(R1)   ;DATA IS MOVED IN
 10445  041310  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 10446  041312  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 10447  041314  000200                         RTS     R0              ;RETURN TO MAIN PROGRAM
 10448
 10449
 10450
```

```
10445                                   ;*THIS SUBROUTINE SETS UP FOR SEARCH
10446                                   ;*CALL IS
10447                                   ;*      JSR     R0,@#SRCH
10448                                   ;*      C               ;CYLINDER
10449                                   ;*.BYTE S               ;SECTOR
10450                                   ;*.BYTE T               ;TRACK
10451
10452  041316  012077  140770          SRCH:   MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
10453  041322  012077  140756                  MOV     (R0)+,@RHDST    ;SET DESIRED SECTOR/TRACK ADDRESS
10454  041326  013777  002434  140744          MOV     @#SERCH,@RHCS1  ;GET READY FOR SEARCH
10455                                                                  ;WITH 30 IN RHCS1
10456  041334  000200                         RTS     R0
10457
10458
10459
10460
10461
10462
10463
10464                                   ;*THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
10465                                   ;*CALL IS
10466                                   ;*      JSR     R0,@#SEEKCY
10467                                   ;*      C               ;CYLINDER
10468                                   ;*
10469
10470  041336  012077  140750          SEEKCY: MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
10471  041342  013777  002452  140730          MOV     @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
10472  041350  000200                         RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
10473
10474                                   ;*THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
10475                                   ;*CALL IS
10476                                   ;*      JSR     R0,@#OFSET
10477                                   ;*      0               ;MICRO INCHES OFSET
10478
10479  041352  052077  140732          OFSET:  BIS     (R0)+,@RHOF     ;SET OFSET REGISTER
10480  041356  013777  002454  140714          MOV     @#OFSETC,@RHCS1 ;MOV14 INTO RHCS1
10481  041364  000200                  RTS     R0              ;RETURN TO MAIN PROGRAM
10482
10483
10484  041366  013701  002300          CLDISK: MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
10485  041372  013702  002276                  MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
10486  041376  013703  002322                  MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
10487  041402  013704  002302                  MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1
10488
10489  041406  012712  000040                  MOV     #CLR,@R2                ;CLEAR ALL REG.
10490  041412  013712  004716                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
10491  041416  005011                  CLR     @R1                     ;CLEAR FUNCTION BITS
10492  041420  000207                  RTS     PC
```

```
10493
10494
10495
10496
10497                                   ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
10498                                   ;*AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY
10499                                   ;* (DRY) IN RHDS1 = 1
10500
10501                                   ;*IT ALSO CHECKS THAT THERE ARE NO BITS STUCK AT 1 IN RHDS1
10502
10503
10504  041422  000000                   PCJSR:  0                       ;PC OF JSR
10505
10506  041424  011637  041422           CHECK:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
10507  041430  162737  000004  041422           SUB     #4,@#PCJSR      ;GET PC OF JSR
10508  041436  011346                           MOV     @R3,-(SP)       ;GET RHDS1
10509  041440  052716  000100                   BIS     #VV,(SP)        ;DONT CHECK VV BIT
10510  041444  000406                           BR      CHECKC          ;GOTO COMMON CHECK ROUTINE
10511
10512  041446  011637  041422           CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
10513  041452  162737  000004  041422           SUB     #4,@#PCJSR      ;GET PC OF JSR
10514  041460  011346                           MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
10515
10516  041462  011146                   CHECKC: MOV     @R1,-(SP)       ;GET CS1
10517  041464  042716  173577                   BIC     #173577,(SP)    ;CLEAR UNWANTED BITS
10518  041470  022726  004200                   CMP     #DVA!RDY,(SP)+  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
10519                                                                   ;AND BE READY
10520  041474  001403                           BEQ     3$              ;BRANCH IF IT DOES
10521  041476  011137  001122                   MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
10522  041502  104061                           ERROR   61              ;RHCS1 DID NOT HAVE DEVICE
10523                                                                   ;AVAILABLE RIGHT AT THE START
10524                                                                   ;ALL OTHER BITS SHOULD BE 0
10525
10526  041504  042716  102000           3$:     BIC     #ATA!LBT,(SP)   ;CLEAR UNWANTED BITS
10527  041510  022726  010700                   CMP     #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
10528  041514  001404                           BEQ     7$              ;BRANCH IF GOOD
10529  041516  011337  001122                   MOV     @R3,@#$BDADR    ;BAD DATA IN  REGISTER (RHDS1)
10530  041522  104062                           ERROR   62              ;RHDS1 HAS SOME BITS OTHER
10531                                                                   ;THAN MOL, DRY, DPR,VV SET
10532                                                                   ;ALL OTHER BITS SHOULD BE 0
10533  041524  000207                           RTS     PC              ;RETURN TO TEST AND HALT/CONTINUE
10534                                                                   ;DEPENDING ON WHETHER THIS IS A
10535                                                                   ;"FATAL" ERROR
10536
10537  041526  062716  000006           7$:     ADD     #6,(SP)         ;ADJUST STACK TO JUMP OVER HALT IN TEST
10538  041532  000207                           RTS     PC              ;RETURN TO THE TEST AND CONTINUE
10539
```

```
10540
10541                                   ;*THIS IS A SUBROUTINE TO SAVE REGISTERS
10542                                   ;*IN THE REGISTER TABLE TO ANY LOCATION
10543                                   ;*THE CALL IS
10544                                   ;*JSR   R0,@#SAVER
10545                           ;*      F       ;FROM
10546                           ;*      T       ;TO
10547                           ;*      N       ;NUMBER OF WORDS SAVED
10548                           ;*F MUST ALWAYS BE RHCS1
10549                           ;*T MUST ALWAYS BE SAVRE
10550
10551   041534                  SAVER:
10552   041534  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10553   041536  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10554   041540  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10555   041542  012001                  MOV     (R0)+,R1        ;FROM
10556   041544  012002                  MOV     (R0)+,R2        ;TO
10557   041546  012003                  MOV     (R0)+,R3        ;NUMBER
10558   041550  013122          1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
10559   041552  005303                  DEC     R3              ;COUNT
10560   041554  001375                  BNE     1$              ;BRANCH IF NOT DONE
10561   041556  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10562   041560  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10563   041562  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10564   041564  000200                  RTS     R0
10565
10566
10567
10568
10569
10570
10571
10572
10573
10574
10575
10576
10577
10578
10579
10580                           ;*WHEN AN EVENT IS TO BE TIMED THE RP04 VECTORS TO "TIME 1"
10581                           ;*PRIORITY OF PROCESS OR IS 4
10582                           ;*PRIORITY OF TRAPS MUST BE 6
10583                           ;*PRIORITY OF RP04 INTERRUPTS IS 7
10584                           ;*
10585
10586   041566  005077  140552  TIME1:  CLR     @PCLCSR                 ;STOP THE CLOCK
10587   041572  017737  140552  041624  MOV     @PCLCTR,@#WAITTM ;GET TIME ON CLOCK
10588   041600  017737  140530  004664  TIME2:  MOV     @RHCC,@#FINACC          ;GET CURRENT CYLINDER
10589   041606  017737  140524  004662          MOV     @RHLA,@#FINALA          ;GET LOOK AHEAD
10590   041614  000002                  RTI                     ;RETURN TO WAIT P OR WAIT.T
10591
10592
10593
```

```
10594                                   ;*THIS IS A WAIT LOOP WHEN AN EVENT IS TO BE TIMED
10595                                   ;*THE CALL IS
10596                                   ;*      WAT
10597                                   ;*      A       ;ABSOLUTE REGISTER ADDRESS
10598                                   ;*      B       ;BIT WAITED FOR
10599                                   ;*      TA      ;TIME ALLOWED GIVEN IN 10 MICROSEC
10600                                   ;*      TO      ;TOLERANCE PLUS/MINUS IN 10 MICROSEC
10601                                   ;*
10602                                   ;*R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
10603                                   ;*R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
10604                                   ;*MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
10605                                   ;*FOR THE SLOWEST PROCESSOR
10606
10607  041616  000000                  WAITPC: 0                       ;WAT PC
10608  041620  000000                  WAITRE: 0                       ;WAIT ON REGISTER ADDRESS
10609  041622  000000                  WAITBT: 0                       ;WAIT ON BIT
10610  041624  000000                  WAITTM: 0                       ;WAITED TIME
10611  041626  005037  041624          WAIT.P: CLR     @#WAITTM        ;CLEAR WAITED TIME
10612  041632  005077  140510                  CLR     @PCLBUF ;CLEAR COUNT SET BUFFER
10613  041636  012777  000021  140500        MOV     #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
10614  041644  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10615  041646  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10616  041650  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10617  041652  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10618  041654  016600  000010                MOV     10(SP),R0       ;R0 HAS ADDRESS OF NEXT LOCATION
10619  041660  010037  041616                MOV     R0,@#WAITPC     ;NOW WAITPC HAS WAT PC + 2
10620  041664  162737  000002  041616        SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
10621  041672  013037  041620                MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
10622  041676  012037  041622                MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
10623  041702  012001                        MOV     (R0)+,R1        ;R1 HAS TIME IN 10 MSEC
10624  041704  012002                        MOV     (R0)+,R2        ;R2 HAS TOLERANCE IN 10 MSEC
10625  041706  010066  000010                MOV     R0,10(SP)       ;RESTORE RETURN ON STACK
10626
10627                                   ;*THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
10628  041712  013703  042064                MOV     @#TIMCNT,R3     ;R3 IS A TEMPORARY COUNTER
10629  041716  033777  041622  177674  1$:   BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
10630  041724  001025                        BNE     4$              ;BRANCH IF YES
10631  041726  005303                        DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
10632  041730  001372                        BNE     1$
10633  041732  013703  042064                MOV     @#TIMCNT,R3     ;TEMPORARY COUNTER
10634  041736  033777  041622  177654  2$:   BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
10635  041744  001015                        BNE     4$              ;BRANCH IF YES
10636  041746  005303                        DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
10637  041750  001372                        BNE     2$
10638  041752  017737  177642  001126        MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
10639  041760  032777  000100  140312        BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
10640  041766  001402                        BEQ     3$              ;BRANCH IF YES
10641  041770  104001                        ERROR   1               ;RP04 DID NOT INTERRUPT
10642  041772  000427                        BR      7$              ;OUT
10643  041774  104002                3$:     ERROR   2               ;RP04 INTERRUPTED BUT WAITED
10644                                                                 ;ON BIT DID NOT OCCUR
10645                                                                 ;EVEN AFTER TWO COUNT DOWNS
10646                                                                 ;FROM 177777 TO 0
10647  041776  000425                        BR      7$              ;OUT
```

```
10648
10649                                          ;*NOW TIME AND TOLERANCE WILL BE CHECKED
10650  042000  017737  177514  001126  4$:     MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
10651  042006  032777  000100  140264          BIT     #IE,@RHCS1       ;DID ANY INTERRUPT OCCUR
10652  042014  001402                          BEQ     5$               ;BRANCH IF YES
10653  042016  104003                          ERROR   3                ;INTERRUPT DID NOT OCCUR EVEN
10654                                                                   ;AFTER ONE BNE AND ONE MOV
10655                                                                   ;OF THE WAITED ON BIT SETTING
10656  042020  000414                          BR      7$               ;OUT
10657  042022  160201                  5$:     SUB     R2,R1            ;R1 NOW HAS LOWER LIMIT OF TIME
10658  042024  023701  041624                  CMP     @#WAITTM,R1      ;FOR GOOD RESULTS, WAITTM
10659                                                                   ;MUST BE GREATER OR EQUAL
10660                                                                   ;TO R1
10661  042030  103002                          BHIS    6$               ;BRANCH IF GOOD
10662  042032  104004                          ERROR   4                ;BIT DID OCCUR BUT TIME
10663                                                                   ;TAKEN IS BELOW LOWER LIMIT
10664  042034  000406                          BR      7$               ;OUT
10665
10666  042036  060202                  6$:     ADD     R2,R2            ;DOUBLE TOLERANCE
10667  042040  060201                          ADD     R2,R1            ;R1 NOW HAS UPPER LIMIT OF TIME
10668  042042  020137  041624                  CMP     R1,@#WAITTM      ;FOR GOOD RESULTS, WAITTM
10669                                                                   ;MUST BE LESS OR EQUAL TO R1
10670  042046  103001                          BHIS    7$               ;BRANCH IF GOOD
10671  042050  104004                          ERROR   4                ;BIT DID OCCUR BUT TIME TAKEN
10672                                                                   ;IS ABOVE UPPER LIMIT
10673  042052                          7$:
10674  042052  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
10675  042054  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
10676  042056  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
10677  042060  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
10678  042062  000002                          RTI                      ;RETURN TO MAIN TEST
10679
10680
10681
10682
10683
10684
10685                                  ;*THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
10686                                  ;*NO TIMING IS DONE
10687                                  ;*CALL IS
10688                                  ;*      WAT
10689                                  ;*      A       ;ABSOLUTE REGISTER ADDRESS
10690                                  ;*      B       ;BIT WAITE) FOR
10691                                  ;*      TA      ;TIME-NOT USED HERE
10692                                  ;*      TO      ;TIME-NOT USED HERE
10693                                  ;*R3-IS A TEMPORARY COUNTER
10694  042064  177777                  TIMCNT: 177777                   ;COUNT FOR WAIT LOOP
10695
10696  042066                          WAIT.T:
10697  042066  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
10698  042070  010346                          MOV     R3,-(SP)         ;;PUSH R3 ON STACK
10699  042072  016600  000004                  MOV     4(SP),R0         ;R0 HAS ADDRESS OF NEXT LOCATION
10700  042076  010037  041616                  MOV     R0,@#WAITPC      ;WAT PC +2 IS IN WAITPC
10701  042102  162737  000002  041616          SUB     #2,@#WAITPC      ;WAT PC IS IN WAITPC
```

```
:0702  042110  013037  041620                    MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
:0703  042114  012037  041622                    MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
:0704  042120  022020                            CMP     (R0)+,(R0)+     ;DUMP NEXT TWO WORDS-TA, TC
:0705  042122  010066  000004                    MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
:0706
:0707                                            ;*THIS HAS THE TWO COUNT DOWNS FROM 177777
:0708  042126  013703  042064                    MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
:0709  042132  033777  041622  177460  1$:       BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
:0710  042140  001025                            BNE     4$              ;BRANCH IF YES
:0711  042142  005303                            DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
:0712  042144  001372                            BNE     1$
:0713  042146  013703  042064                    MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
:0714  042152  033777  041622  177440  2$:       BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
:0715  042160  001015                            BNE     4$              ;BRANCH IF YES
:0716  042162  005303                            DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
:0717  042164  001372                            BNE     2$
:0718  042166  017737  177426  001126            MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
:0719  042174  032777  000100  140076            BIT     #IE.DRHCS1      ;DID ANY INTERRUPT OCCUR
:0720  042202  001402                            BEQ     3$              ;BRANCH IF YES
:0721  042204  104001                            ERROR   1               ;RP04 DID NOT INTERRUPT
:0722                                                                    ;BIT DID NOT OCCUR
:0723  042206  000414                            BR      5$              ;OUT
:0724  042210  104002                  3$:       ERROR   2               ;RP04 INTERRUPTED BUT
:0725                                                                    ;WAITED ON BIT DID NOT OCCUR
:0726                                                                    ;EVEN AFTER TWO COUNT DOWNS
:0727                                                                    ;FROM 177777 TO 0
:0728  042212  000412                            BR      5$              ;OUT
:0729
:0730                                            ;*BIT DID SET SO CHECK IF INTERRUPT OCCURED
:0731  042214  000240                  4$:       NOP                     ;ALLOW TIME FOR INTERRJPT
:0732  042216  032777  000100  140054            BIT     #IE.DRHCS1      ;DID ANY INTERRUPT OCCUR
:0733  042224  001405                            BEQ     5$              ;BRANCH IF YES
:0734  042226  017737  177366  001126            MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
:0735  042234  104003                            ERROR   3               ;INTERRUPT DID NOT OCCUR
:0736                                                                    ;EVEN AFTER ONE BNE OF
:0737                                                                    ;THE WAITED ON BIT OCCURING
:0738  042236  000400                            BR      5$              ;OUT
:0739  042240                           5$:
:0740  042240  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
:0741  042242  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
:0742  042244  000002                            RTI                     ;RETURN TO MAIN TEST
```

```
                                        ;*THIS CHANGES REGISTER SAVED VALUE
                                        ;*CALL IS
                                        ;*      JSR     R0,J#CHREG
                                        ;*      R                       ;REGISTER TO BE CHANGED
                                        ;*      N                       ;NUMBER OF BITS TO BE CHANGED
                                        ;*      NEW                     ;NEW VALUE OF BIT MUST BE C OR 1
                                        ;*      P                       ;POSITION OF BIT TO BE CHANGED
                                        ;*NEW AND P WILL BE REPEATED N NUMBER OF TIMES
042246                                  CHREG:
042246  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042250  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042252  012001                                  MOV     (R0)+,R1        ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
042254  012002                                  MOV     (R0)+,R2        ;R2 HAS NUMBER OF CHANGES
042256  162701  002272                        SUB     #RHWC,R1        ;R1 HAS OFSET OF REQUIRED REGISTER
042262  005720                        1$:     TST     (R0)+           ;IS A BIC OR A BIS TO BE DONE
042264  001403                                  BEQ     2$              ;BRANCH IF A BIC IS REQUIRED
042266  052061  004612                        BIS     (R0)+,SAVERE(R1) ;SET REQUIRED BIT
042272  000402                                  BR      3$              ;BRANCH TO DECREMENT COUNT
042274  042061  004612                2$:     BIC     (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
042300  005302                        3$:     DEC     R2              ;DECREMENT NUMBER OF CHANGES
042302  001367                                  BNE     1$              ;BRANCH IF NOT COMPLETE
042304  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
042306  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
042310  000230                                  RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
                                        ;*THIS FILLS A BLOCK WITH INCREMENTAL DATA
                                        ;*CALL IS
                                        ;*      JSR     R0,J#FILL
                                        ;*      F                       ;FROM
                                        ;*      N                       ;NUMBER OF WORDS
                                        ;*      S                       ;STARTING VALUE OF DATA
                                        ;*      I                       ;INCREMENT DATA BY
042312                                  FILL:
042312  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042314  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042316  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042320  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
042322  012001                                  MOV     (R0)+,R1        ;R1 HAS ADDRESS WHERE DATA IS TO GO
042324  012002                                  MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS TO BE FILLED
042326  012003                                  MOV     (R0)+,R3        ;STARTING VALUE OF DATA
042330  012004                                  MOV     (R0)+,R4        ;R4 HAS INCREMENT
                                        ;*NOW DATA WILL BE FILLED
042332  010321                        1$:     MOV     R3,(R1)+        ;FILL DATA
042334  060403                                ADD     R4,R3           ;GET NEXT VALUE OF DATA
042336  005302                                DEC     R2              ;DECREMENT COUNT
042340  001374                                BNE     1$              ;BRANCH IF ALL NOT DONE
042342  012604                                MOV     (SP)+,R4        ;;POP STACK INTO R4
042344  012603                                MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
 :007F7  042346  012602                 MOV     (SP)+,R2        ;;POP STACK INTO R2
 :00800  042350  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
 :00804  042352  000205                 RTS     R0              ;RETURN TO MAIN PROGRAM
 :00806
 :00810
 :00812
 :00814
```

```
 10803
 10804
 10805
 10806                                  ;*THIS IS A SUBROUTINE TO COMPARE REGISTERS
 10807                                  ;*GOOD DATA IS ALREADY SAVED IN 'SAVERE'
 10808                                  ;*TEST DATA IS IN THE REGISTERS
 10809                                  ;*CALL IS
 10810                                  ;*      JSR     R0,@#COMREG
 10811                                  ;*      SAVERE                  ;GOOD DATA
 10812                                  ;*      RHCS1                   ;ADDRESS OF ADDRESS TEST DATA
 10813                                  ;*      N.                      ;RETURN FOR ERROR
 10814                                  ;*      RG                      ;RETURN FOR GOOD COMPARISON
 10815                                  ;*ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
 10816                                  ;*'REGADR' HAS REGISTER ADDRESS
 10817
 10818  042354                         COMREG:
 10819  042354  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 10820  042356  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 10821  042360  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 10822  042362  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 10823  042364  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 10824  042366  012001                         MOV     (R0)+,R1        ;R1 HAS ADDRESS OF GOOD DATA
 10825  042370  012002                         MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
 10826  042372  012003                         MOV     (R0)+,R3        ;R3 HAS NUMBER OF WORDS
 10827  042374  012004                         MOV     (R0)+,R4        ;R4 HAS RETURN FOR ERROR
 10828  042376  011000                         MOV     (R0),R0         ;R0 HAS RETURN ON NO ERROR
 10829                                  ;*NOW SAVE REGISTERS
 10830  042400  004737  043300                 JSR     PC,@#PUTREG     ;SAVE REGISTERS
 10831  042404  113737  004637  002401         MOVB    @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF RHAS SAME
 10832  042412  012705  177776                 MOV     #-2,R5          ;PRESET R5 TO -2
 10833                                  ;*NOW COMPARES WILL MADE
 10834  042416  062705  000002         1$:     ADD     #2,R5           ;INCREMENT TO INDEX
 10835  042422  022122                         CMP     (R1)+,(R2)+     ;COMPARE REGISTER CONTENTS
 10836  042424  001420                         BEQ     2$              ;BRANCH IF GOOD
 10837  042426  014137  001124                 MOV     -(R1),@#$GDDAT  ;SAVE GOOD DATA
 10838  042432  014237  001126                 MOV     -(R2),@#$BDDAT  ;SAVE BAD DATA
 10839  042436  016537  002272  004600         MOV     RHWC(R5),@#REGADR       ;SAVE ADDRESS OF FAILING REGISTER
 10840  042444  004714                         JSR     PC,@R4          ;RETURN TO MAIN PROGRAM
 10841                                                                  ;TO PRINT ERROR
 10842  042446  022122                         CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
 10843  042450  017746  136464                 MOV     @SWR,-(SP)      ;GET SWITCH SETTING
 10844  042454  042716  177177                 BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 9
 10845  042460  022726  000200                 CMP     #SW07,(SP)+     ;IS 7 SET AND 9 DOWN
 10846  042464  001402                         BEQ     3$              ;BRANCH OUT IF YES
 10847  042466  005303                 2$:     DEC     R3              ;ARE ALL COMPARES DONE
 10848  042470  001352                         BNE     1$              ;BRANCH IF NOT COMPLETE
 10849
 10850  042472                         3$:
 10851  042472  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
 10852  042474  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
 10853  042476  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
 10854  042500  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 10855  042502  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 10856  042504  000200                         RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
:0857  042506  000000          4$:     .WORD   0               ;TEMP STORAGE
```

```
10858
10859
10860
10861                                   ;*HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
10862                                   ;*ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
10863                                   ;*PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
10864
10865                                   ;*WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
10866                                   ;*THE PROGRAM GOES BACK TO CAN BE CHANGED.
10867                                   ;*THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
10868                                   ;*1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
10869                                   ;*2.  LOOP ON ERROR SWITCH MUST BE SET
10870                                   ;*3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
10871                                   ;*IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
10872                                   ;*THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
10873                                   ;*TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
10874                                   ;*THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
10875                                   ;*COMES TO THE END OF THE TEST UNDER CONSIDERATION.
10876                                   ;*
10877                                   ;*AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
10878                                   ;*NORMAL OPERATION WILL CONTINUE.
10879
10880  042510  000000                  TESTAD: 0                       ;FIRST ADDRESS OF TEST
10881  042512                          OPERSEL:
10882  042512  005037  177776                  CLR     PS             ;MAKE PROCESSOR STATUS ZERO
10883  042516  012737  177777  046760          MOV     #-1,@#PRITEM   ;CLEAR PREVIOUS ITEM NUMBER
10884  042524  104400  042532                  TYPE    ,65$           ;;TYPE ASCIZ STRING
10885  042530  000421                          BR      64$            ;;GET OVER THE ASCIZ
10886                                  ;;65$:  .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
10887  042574                          64$:
10888  042574  013746  004604                  MOV     @#TSTNM,-(SP)  ;GET READY TO TYPE TEST
10889  042600  104401                          TYPOC                  ;NUMBER
10890  042602  104400  042610                  TYPE    ,67$           ;;TYPE ASCIZ STRING
10891  042606  000414                          BR      66$            ;;GET OVER THE ASCIZ
10892                                  ;;67$:  .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
10893  042640                          66$:
10894  042640  013746  001110                  MOV     @#$LPERR,-(SP) ;GET READY TO TYPE LOOP BACK PC
10895  042644  104401                          TYPOC
10896  042646  104400  001223                  TYPE    ,$CRLF
10897  042652  104400  042660                  TYPE    ,69$           ;;TYPE ASCIZ STRING
10898  042656  000430                          BR      68$            ;;GET OVER THE ASCIZ
10899                                  ;;69$:  .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST
10900  042740                          68$:
10901  042740  104400  042746                  TYPE    ,71$           ;;TYPE ASCIZ STRING
10902  042744  000430                          BR      70$            ;;GET OVER THE ASCIZ
10903                                  ;;71$:  .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON
10904  043026                          70$:
10905  043026  104400  043034                  TYPE    ,73$           ;;TYPE ASCIZ STRING
10906  043032  000422                          BR      72$            ;;GET OVER THE ASCIZ
10907                                  ;;73$:  .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN  <15><12>
10908  043100                          72$:
10909  043100  104411                          RDOCT
10910  043102  062716  000002                  ADD     #2,(SP)        ;GET LPADR
10911  043106  012637  001106                  MOV     (SP)+,@#$LPADR
```

```
 10912  043112  104400  043120                 TYPE    ,75$            ;;TYPE ASCIZ STRING
 10913  043116  000417                         BR      74$             ;;GET OVER THE ASCIZ
 10914                                  ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
 10915  043156                          74$:
 10916  043156  104400  043164                 TYPE    ,77$            ;;TYPE ASCIZ STRING
 10917  043162  000440                         BR      76$             ;;GET OVER THE ASCIZ
 10918                                  ;;77$:  .ASCIZ  <15><12>  THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15>
 10919  043264                          76$:
 10920  043264  104411                         RDOCT
 10921  043266  013637  001110                 MOV     (SP)+,@#$LPERR  ;GET LPERR
 10922  043272  013746  001106                 MOV     @#$LPADR,-(SP)
 10923  043276  000002                         RTI
 10924
 10925
```

```
10926
10927
10928                                   ;*THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
10929                                   ;*IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
10930
10931                                   ;*THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
10932                                   ;*AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
10933                                   ;*ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
10934
10935
10936
10937
10938  043300                   PUTREG:
10939  043300  010046                   MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10940  043302  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10941  043304  010246                   MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10942  043306  012700  002272           MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
10943  043312  012701  002354           MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
10944  043316  012702  000022           MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
10945  043322  013021             10$:  MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
10946  043324  005302                   DEC     R2
10947  043326  001375                   BNE     10$
10948  043330  012602                   MOV     (SP)+,R2        ;;POP STACK INTO R2
10949  043332  012601                   MOV     (SP)+,R1        ;;POP STACK INTO R1
10950  043334  012600                   MOV     (SP)+,R0        ;;POP STACK INTO R0
10951  043336  000207                   RTS     PC
10952
10953
10954
10955                                   ;*THIS IS A DATA  COMMAND SETUP SUBROUTINE
10956                                   ;*THE CALL IS
10957                                   ;*      JSR     R0,@#RUN
10958                                   ;*      C                       ;CYLINDER
10959                                   ;*.BYTE S                       ;SECTOR
10960                                   ;*.BYTE T                       ;TRACK
10961                                   ;*      -W                      ;WORD COUNT
10962                                   ;*      B                       ;BUS ADDRESS
10963                                   ;*      BAI                     ;BUS ADDRESS INHIBIT
10964                                   ;*      FMT22!ECI!HCI           ;FMT22=1 =16 BIT WORDS
10965                                   ;*                              ;ECI = ECC CORRECTION INHIBIT
10966                                   ;*                              ;HCI = HEADER COMPARE INHIBIT
10967                                   ;*      COM                     ;COMMAND ADDRESS
10968  043340  012077  136746   RUN:    MOV     (R0)+,@RHCA     ;CYLINDER
10969  043344  012077  136734           MOV     (R0)+,@RHDST    ;DESIRED SECTOR/TRACK
10970  043350  012077  136716           MOV     (R0)+,@RHWC     ;WORD COUNT
10971  043354  012077  136714           MOV     (R0)+,@RHBA     ;BUS ADDRESS
10972  043360  013746  004716           MOV     @#UNIT,-(SP)    ;GET UNIT NO
10973  043364  052016                   BIS     (R0)+,(SP)      ;SET BUS ADDRESS INHIBIT
10974  043366  012677  136704           MOV     (SP)+,@RHCS2    ;UNIT NO AND BAI TO RHCS2
10975  043372  012077  136712           MOV     (R0)+,@RHOF     ;FORMAT, ECC INHIBIT, HEADER
10976                                                           ;COMPARE, IF THERE
10977  043376  013077  136676           MOV     @(R0)+,@RHCS1   ;COMMAND IN RHCS1
10978  043402  000200                   RTS     R0              ;RETURN TO MAIN PROGRAM
10979
```

```
 10990
```

```
10981
10982
10983                                   ;*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
10984
10985                                   ;*R1 HAS GOOD DATA BUFFER ADDRESS
10986                                   ;*R2 HAS TEST DATA BUFFER ADDRESS
10987                                   ;*R5 HAS ADDRESS OF RETURN ON ERROR
10988                                   ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
10989                                   ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
10990
10991                                   ;*CALL IS:
10992                                   ;*      JSR     R0,@#COMPAR
10993                                   ;*      G                       ;ADDRESS OF GOOD DATA
10994                                   ;*      T                       ;ADDRESS OF TEST DATA
10995                                   ;*      N                       ;NUMBER OF WORDS TO BE COMPARED
10996                                   ;*      RE                      ;RETURN ON ERROR
10997                                   ;*      RG                      ;RETURN ON NO ERROR
10998
10999
11000
11001   043404                          COMPAR:
11002   043404  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11003   043406  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11004   043410  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11005   043412  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11006   043414  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11007   043416  012001                          MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
11008   043420  012002                          MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
11009   043422  012003                          MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
11010   043424  012005                          MOV     (R0)+,R5        ;RETURN ON ERROR
11011   043426  011000                          MOV     (R0),R0         ;RETURN ON NO ERROR
11012   043430  010304                          MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
11013   043432  005204                          INC     R4
11014   043434  010437  004602          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
11015   043440  022122                          CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
11016   043442  001417                          BEQ     2$              ;BRANCH IF GOOD
11017
11018   043444  014137  001124                  MOV     -(R1),@#$GDDAT  ;GOOD DATA
11019   043450  014237  001126                  MOV     -(R2),@#$BDDAT  ;BAD DATA
11020   043454  160337  004602                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
11021   043460  004715                          JSR     PC,@R5          ;RETURN TO PRINT ERROR
11022   043462  022122                          CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
11023   043464  017746  135450                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
11024   043470  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
11025   043474  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
11026   043500  001402                          BEQ     3$              ;BRANCH OUT IF YES
11027   043502  005303                  2$:     DEC     R3              ;COUNT
11028   043504  001353                          BNE     1$              ;BRANCH IF ALL NOT DEVICE
11029   043506                          3$:
11030   043506  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11031   043510  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11032   043512  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11033   043514  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11034   043516  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
 11035  043520  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
11036
11037
11038                                   ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
11039                                   ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
11040
11041  043522                           BASECH:
11042  043522  104400  043530                   TYPE    ,65$            ;;TYPE ASCIZ STRING
11043  043526  000425                           BR      64$             ;;GET OVER THE ASCIZ
11044                                   ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS   /
11045  043602                           64$:
11046  043602  013746  002300                   MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
11047  043606  104401                           TYPOC
11048  043610  104400  043616                   TYPE    ,67$            ;;TYPE ASCIZ STRING
11049  043614  000425                           BR      66$             ;;GET OVER THE ASCIZ
11050                                   ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
11051  043670                           66$:
11052  043670  004737  045352                   JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
11053  043674  104411                           RDOCT
11054  043676  012700  002270                   MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
11055  043702  012701  000024                   MOV     #20.,R1         ;NUMBER OF REGISTERS
11056  043706  042710  177700           1$:     BIC     #^C77,(R0)      ;CLEAR OLD BASE
11057  043712  051620                           BIS     (SP),(R0)+      ;SET NEW BASE
11058  043714  005301                           DEC     R1              ;COUNT
11059  043716  001373                           BNE     1$              ;BRANCH IF 20 NOT DONE
11060  043720  104400  043726                   TYPE    ,69$            ;;TYPE ASCIZ STRING
11061  043724  000417                           BR      68$             ;;GET OVER THE ASCIZ
11062                                   ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
11063  043764                           68$:
11064  043764  013746  002266                   MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
11065  043770  104401                           TYPOC
11066  043772  104400  044000                   TYPE    ,71$            ;;TYPE ASCIZ STRING
11067  043776  000437                           BR      70$             ;;GET OVER THE ASCIZ
11068                                   ;;71$:  .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
11069  044076                           70$:
11070  044076  104411                           RDOCT
11071  044100  012637  002266                   MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
11072  044104  104400  044112                   TYPE    ,73$            ;;TYPE ASCIZ STRING
11073  044110  000421                           BR      72$             ;;GET OVER THE ASCIZ
11074                                   ;;73$:  .ASCIZ  <15><12>/RESTART PROGRAM FROM 200 OR 210/
11075  044154                           72$:
11076  044154  104400  044162                   TYPE    ,75$            ;;TYPE ASCIZ STRING
11077  044160  000416                           BR      74$             ;;GET OVER THE ASCIZ
11078                                   ;;75$:  .ASCIZ  <15><12>/NEW BASE WILL REMAIN   - /
11079  044216                           74$:
11080  044216  013746  002300                   MOV     @#RHCS1,-(SP)
11081  044222  104401                           TYPOC
11082  044224  104400  044232                   TYPE    ,77$            ;;TYPE ASCIZ STRING
11083  044230  000416                           BR      76$             ;;GET OVER THE ASCIZ
11084                                   ;;77$:  .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN - /
11085  044266                           76$:
11086  044266  013746  002266                   MOV     @#RPVEC,-(SP)
11087  044272  104401                           TYPOC
11088  044274  104400  044302                   TYPE    ,79$            ;;TYPE ASCIZ STRING
11089  044300  000402                           BR      78$             ;;GET OVER THE ASCIZ
```

```
11090                                          ;;79$:  .ASCIZ  <15><12>/ /
11091  044306                                  78$:
11092  044306  104400  044314                         TYPE    81$             ;;TYPE ASCIZ STRING
11093  044312  000416                                 BR      80$             ;;GET OVER THE ASCIZ
11094                                          ;;81$:  .ASCIZ  /15><12>/UNTIL PROGRAM IS RELOADED/
11095  044350                                  80$:
11096  044350  000000                                 HALT
11097
11099
```

```
                                        ;;**********************************************************************
                                        ;
044352                                  RPVECT:
044352  104430  044360                          TYPE    ,65$                    ;;TYPE ASCIZ STRING
044356  000411                                  BR      64$                     ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  /TRAPED FROM PC = /
044402                                  64$:
044402  104401                                  TYPOC                           ;TYPE FROM PC
044404  012737  044352  135654                  MOV     #RPVECT,@#RPVEC         ;RESTORE TRAP RP04 VECTOR
044412  000000                                  HALT                            ;CHANGE TO CONTINUE
                                        ;;**********************************************************************
```

```
                                        .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

044414                                  $SCOPE:
044414  104406                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
044416  032777  040000  134514          1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
044424  001111                                  BNE     $OVER           ;;YES IF SW14=1
                                        ;*****START OF CODE FOR THE XOR TESTER*****
044426  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
044430  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
044434  012737  044454  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
044442  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
044446  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
044452  000463                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
044454  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
044456  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
044462  000423                                  BR      7$              ;;LOOP ON THE PRESENT TEST
044464                                  6$:;*****END OF CODE FOR THE XOR TESTER*****
044464  032777  000400  134446                  BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
044472  001404                                  BEQ     2$              ;;BR IF NO
044474  127737  134440  001102                  CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
044502  001462                                  BEQ     $OVER           ;;BR IF YES
044504  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
044510  001421                                  BEQ     3$              ;;BR IF NO
044512  123737  001115  001103                  CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
044520  101015                                  BHI     3$              ;;BR IF NO
044522  032777  001000  134410                  BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
044530  001404                                  BEQ     4$              ;;BR IF NO
044532  013737  001110  001106          7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
044540  000443                                  BR      $OVER
044542  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
044546  005037  001212                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
044552  000415                                  BR      1$              ;;ESCAPE TO THE NEXT TEST
044554  032777  004000  134356          3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
044562  001011                                  BNE     1$              ;;BR IF YES
044564  005737  001100                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
044570  001406                                  BEQ     1$              ;;      INHIBIT ITERATIONS
044572  005237  001104                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
044576  023737  001212  001104                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
044604  002021                                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
044606  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
044614  013737  044664  001212                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
```

```
 ;;168  044622  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 ;;169  044626  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 ;;170  044632  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 ;;171  044636  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 ;;172  044642  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 ;;173  044650  013777  001102  134264  $SVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 ;;174  044656  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 ;;175  044662  000002                          RTI                     ;;FIXES PS
 ;;176  044664  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

                                        $TYPDS:
                                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
                                                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
                                                BPL     1$              ;;BR IF INPUT IS POS.
                                                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
                                                MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
                                        1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
                                                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
                                                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
                                        2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
                                        3$:     MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
                                                SUB     R1,R5           ;;FORM THIS BCD DIGIT
                                                BLT     4$              ;;BR IF DONE
                                                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
                                                BR      3$
                                        4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
                                                TST     R2              ;;CHECK IF BCD DIGIT=0
                                                BNE     5$              ;;FALL THROUGH IF 0
                                                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
                                                BMI     7$              ;;BR IF YES
                                        5$:     ASLB    (SP)            ;;MSD?
                                                BCC     6$              ;;BR IF NO
                                                MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
                                        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
                                        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT A READY A DIGIT
                                                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                                                TST     (R0)+           ;;JUST INCREMENTING
                                                CMP     R0,#10          ;;CHECK THE TABLE INDEX
                                                BLT     2$              ;;GO DO THE NEXT DIGIT
                                                BGT     8$              ;;GO TO EXIT
                                                MOV     R5,R2           ;;GET THE LSD
                                                BR      6$              ;;GO CHANGE TO ASCII
                                        8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                                                BPL     9$              ;;BR IF NO
                                                MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                                        9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                                                MOV     (SP)+,R5        ;;POP STACK INTO R5
```

F08


```
011231  045044  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
011232  045046  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
011233  045050  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
011234  045052  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
011235  045054  104400  045103                  TYPE    $DBLK           ;;NOW TYPE THE NUMBER
011236  045060  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
011237  045066  012616                          MOV     (SP)+,(SP)
011238  045070  000002                          RTI                     ;;RETURN TO USER
011239  045072  023420                  $DTBL:  10000.
011240  045074  001750                          1000.
011241  045076  000144                          100.
011242  045100  000012                          10.
011243  045102  000004                  $DBLK:  .BLKW   4
```

```
                                .SBTTL  TYPE ROUTINE

                                ;;**************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    .MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

045112  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
045116  100002                          BPL     1$              ;;BR IF YES
045120  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
045122  000407                          BR      3$              ;;LEAVE
045124  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
045126  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
045132  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
045134  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
045136  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
045140  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
045142  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
045146  000002                          RTI                     ;;RETURN
045150  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
045154  001430                          BEQ     8$
045156  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
045162  001006                          BNE     5$
045164  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
045166  104400                          TYPE                    ;;TYPE A CR AND LF
045170  001223                          $CRLF
045172  105037  045326                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
045176  000755                          BR      2$              ;;GET NEXT CHARACTER
045200  004737  045262          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
045204  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
045210  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
045212  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
045216  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
045222  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
045224  004737  045262                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
045230  105337  045326                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
045234  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

045236  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
045242  004737  045262          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
045246  132737  000007  045326          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
```

```
11298  045254  001372                          BNE     9$              ::TAB STOP
11299  045256  005726                          TST     (SP)+           ::POP SPACE OFF STACK
11300  045260  000724                          BR      2$              ::GET NEXT CHARACTER
11301  045262  105777  133662          $TYPEC: TSTB    @$TPS           ::WAIT UNTIL PRINTER IS READY
11302  045266  100375                          BPL     $TYPEC
11303  045270  116677  000002  133654          MOVB    2(SP),@$TPB     ::LOAD CHAR TO BE TYPED INTO DATA REG.
11304  045276  122766  000015  000002          CMPB    #CR,2(SP)       ::IS CHARACTER A CARRIAGE RETURN?
11305  045304  001003                          BNE     1$              ::BRANCH IF NO
11306  045306  105037  045326                  CLRB    $CHARCNT        ::YES--CLEAR CHARACTER COUNT
11307  045312  000406                          BR      $TYPEX          ::EXIT
11308  045314  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ::IS CHARACTER A LINE FEED?
11309  045322  001402                          BEQ     $TYPEX          ::BRANCH IF YES
11310  045324  105227                          INCB    (PC)+           ::COUNT THE CHARACTER
11311  045326  000000                  $CHARCNT:.WORD  0               ::CHARACTER COUNT STORAGE
11312  045330  000207                  $TYPEX: RTS     PC
11313
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
045332  000000                          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
045334  000000                          $TKQIN: .WORD   0               ;;INPUT POINTER
045336  000000                          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
045340  000011                          $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
        045351                          $TKQEND=.
        045352                          .EVEN

                                        ;*TK INITIALIZE ROUTINE
                                        ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                        ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                        ;
                                        ;*CALL:
                                        ;*      JSR     PC,$TKINT
                                        ;*      RETURN
                                        ;
045352  005037  045332                  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
045356  012737  045340  045334                  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
045364  013737  045334  045336                  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
045372  012737  045422  000060                  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
045400  012737  000200  000062                  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
045406  005777  133534                        TST     @$TKB           ;;CLEAR DONE FLAG
045412  012777  000100  133524                MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
045420  000207                              RTS     PC              ;;RETURN TO CALLER

                                        ;*TK SERVICE ROUTINE
                                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                        ;*IT IN THE QUEUE.
                                        ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
                                        ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL
                                        ;
045422  117746  133520                  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
045426  042716  177600                          BIC     #^C177,(SP)     ;;STRIP THE JUNK
045432  021627  000003                          CMP     (SP),#3         ;;IS IT A CONTROL C?
045436  001007                                BNE     1$              ;;BRANCH IF NO
045440  104400  046411                        TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
045444  004737  045352                        JSR     PC,$TKINT       ;;INIT THE KEYBOARD
045450  005726                              TST     (SP)+           ;;CLEAN UP STACK
045452  000137  042512                        JMP     OPERSEL         ;;CONTROL C RESTART
045456  021627  000007                  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
045462  001004                                BNE     2$              ;;BRANCH IF NO
045464  022737  000176  001140                CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
045472  001500                                BEQ     6$              ;;GO TO SWR CHANGE

045474                                  2$:
045474  022737  000011  045332                CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
045502  001004                                BNE     3$              ;;BRANCH IF NO
045504  104400  001216                        TYPE    ,$BELL          ;;RING THE TTY BELL
045510  005726                              TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
045512  000451                              BR      5$              ;;EXIT
```

```
11368  045514  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
11369  045520  001021                          BNE     32$             ;;BRANCH IF NO
11370  045522  005077  133416                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11371  045526  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
11372  045530  105777  133410          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
11373  045534  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
11374  045536  117746  133404                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
11375  045542  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11376  045546  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11377  045552  001366                          BNE     31$             ;;BRANCH IF NO
11378  045554  012777  000100  133362          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
11379  045562  000002                          RTI                     ;;RETURN
11380  045564  005237  045332          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
11381  045570  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
11382  045574  002405                          BLT     4$              ;;BRANCH IF YES
11383  045576  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
11384  045602  003002                          BGT     4$              ;;BRANCH IF YES
11385  045604  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
11386  045610  112677  177520          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
11387  045614  005237  045334                  INC     $TKQIN          ;;UPDATE THE POINTER
11388  045620  023727  045334  045351          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
11389  045626  001003                          BNE     5$              ;;BRANCH IF NO
11390  045630  012737  045340  045334          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
11391  045636  000002                  5$:     RTI                     ;;RETURN
11392
11393                                          ;;**********************************************************************
11394                                          ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11395                                          ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
11396                                          ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11397                                          ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11398  045640  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
11399  045646  001124                          BNE     15$             ;;EXIT IF NOT
11400  045650  105777  133270                  TSTB    @$TKS           ;;IS A CHAR WAITING?
11401  045654  100121                          BPL     15$             ;;IF NOT, EXIT
11402  045656  117746  133264                  MOVB    @$TKB,-(SP)     ;;YES
11403  045662  042716  177600                  BIC     #^C17?,(SP)     ;;MAKE IT  7-BIT ASCII
11404  045666  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
11405  045672  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
11406                                                                  ;;AND EXIT
11407
11408                                          ;;**********************************************************************
11409                                          ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11410                                          ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11411                                          ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11412  045674  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11413  045702  001674                          BEQ     2$              ;;BRANCH IF YES
11414  045704  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
11415  045706  004737  045352                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
11416  045712  005077  133226                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11417  045716  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
11418
11419  045724  104400  046423                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
11420  045730  104400  046430          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
11421  045734  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
```

```
11422  045740  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11423  045742  104400  046441                TYPE    .$MNEW          ;;PROMPT FOR NEW SWR
11424  045746  005046                19$:     CLR     -(SP)           ;;CLEAR COUNTER
11425  045750  005046                        CLR     -(SP)           ;;THE NEW SWR
11426  045752  105777  133166        7$:     TSTB    @$TKS           ;;CHAR THERE?
11427  045756  100375                        BPL     7$              ;;IF NOT TRY AGAIN
11428
11429  045760  117746  133162                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
11430  045764  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11431
11432  045770  021627  000003                CMP     (SP),#3         ;;IS IT A CONTROL-C?
11433  045774  001015                        BNE     9$              ;;BRANCH IF NOT
11434  045776  104400  046411                TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
11435  046002  062706  000006                ADD     #6,SP           ;;CLEAN UP STACK
11436  046006  123727  001135  000001        CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
11437  046014  001003                        BNE     8$              ;;BRANCH IF NO
11438  046016  012777  000100  133120        MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
11439  046024  000137  042512        8$:     JMP     OPERSEL         ;;CONTROL-C RESTART
11440
11441
11442  046030  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
11443  046034  001005                        BNE     10$             ;;BRANCH IF NOT
11444  046036  104400  046416                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
11445  046042  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11446  046046  000737                        BR      19$             ;;LET'S TRY IT AGAIN
11447
11448
11449  046050  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
11450  046054  001022                        BNE     16$             ;;BRANCH IF NO
11451  046056  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11452  046062  001403                        BEQ     11$             ;;BRANCH IF YES
11453  046064  016677  000002  133046        MOV     2(SP),@SWR      ;;SAVE NEW SWR
11454  046072  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
11455  046076  104400  001223        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
11456  046102  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11457  046110  001003                        BNE     15$             ;;BRANCH IF NOT
11458  046112  012777  000100  133024        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
11459  046120  000002                15$:    RTI                     ;;RETURN
11460  046122  004737  045262        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
11461  046126  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
11462  046132  002420                        BLT     18$             ;;BRANCH IF YES
11463  046134  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
11464  046140  003015                        BGT     18$             ;;BRANCH IF YES
11465  046142  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11466  046146  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
11467  046152  001403                        BEQ     17$             ;;BRANCH IF YES
11468  046154  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
11469  046156  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
11470  046160  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
11471  046162  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
11472  046166  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11473  046172  000667                        BR      7$              ;;GET THE NEXT ONE
11474  046174  104400  001222        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
11475  046200  000720                        BR      20$             ;;SIMULATE CONTROL-U
```

```
11476                                   .DSABL  LSB
11477
11478
11479                                   ;;*****************************************************************
11480                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11481                                   ;*CALL:
11482                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
11483                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11484                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
11485                                   ;
11486
11487  046202  011646                   $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
11488  046204  016666  000004  000002           MOV     4(SP),2(SP)     ;;THE PS
11489  046212  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
11490  046216  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
11491  046220  012746  046226                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
11492  046224  000002                          RTI                     ;;POP NEW PC AND PS
11493  046226                           64$:
11494  046226  005737  045332          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
11495  046232  001775                          BEQ     1$
11496  046234  005337  045332                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
11497  046240  117746  177072  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
11498  046246  005237  045336                  INC     $TKQOUT         ;;UPDATE THE POINTER
11499  046252  023727  045336  045351          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
11500  046260  001003                          BNE     2$              ;;BRANCH IF NO
11501  046262  012737  045340  045336          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
11502  046270  000002                  2$:     RTI                     ;;RETURN
11503                                   ;;*****************************************************************
11504                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11505                                   ;*CALL:
11506                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
11507                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11508                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
11509
11510  046272  010346                   $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
11511  046274  012703  046400          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
11512  046300  022703  046411          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
11513  046304  101405                          BLOS    4$              ;;BR IF YES
11514  046306  104407                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
11515  046310  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
11516  046312  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
11517  046316  001003                          BNE     3$              ;;SKIP IF NOT
11518  046320  104400  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
11519  046324  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
11520  046326  111337  046376          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
11521  046332  104400  046376                  TYPE    ,9$
11522  046336  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
11523  046342  001356                          BNE     2$              ;;LOOP IF NOT RETURN
11524  046344  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
11525  046350  104400  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
11526  046354  012603                          MOV     (SP)+,R3        ;;RESTORE R3
11527  046356  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11528  046360  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
11529  046366  012766  046400  000004          MOV     #$TTYIN,4(SP)
```

```
11530  046374  000002                          RTI                     ;;RETURN
11531  046376     000          9$:              BYTE    0              ;;STORAGE FOR ASCII CHAR. TO TYPE
11532  046377     000                           BYTE    0              ;;TERMINATOR
11533  046400  000011          $TTYIN:          .BLKB   9.             ;;RESERVE 9. BYTES FOR TTY INPUT
11534  046411     136  006503  000012  $CNTLC:  .ASCIZ  /↑C/<15><12>   ;;CONTROL "C"
11535  046416  052536  005015     000  $CNTLU:  .ASCIZ  /↑U/<15><12>   ;;CONTROL "U"
11536  046423     136  006507  000012  $CNTLG:  .ASCIZ  /↑G/<15><12>   ;;CONTROL "G"
11537  046430  005015  053523  020122  $MSWR:   .ASCIZ  <15><12>/SWR = /
11538  046436  020075     000
11539  046441     040  047040  053505  $MNEW:   .ASCIZ  /  NEW = /
11540  046446  036440  000040
11541                                                  ;FROM THE TTY
```

```
11542                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
11543
11544                                   ;;**********************************************************************
11545                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11546                                   ;*CHANGE IT TO BINARY.
11547                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
11548                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11549                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11550                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11551                                   ;*CALL:
11552                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
11553                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
11554                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
11555
11556  046452  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
11557  046454  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
11558  046462  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11559  046464  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11560  046466  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11561  046470  104410                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
11562  046472  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
11563  046474  010037  046600                  MOV     R0,5$           ;;AND SAVE IT
11564  046500  005001                          CLR     R1              ;;CLEAR DATA WORD
11565  046502  005002                          CLR     R2
11566  046504  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
11567  046506  001420                          BEQ     3$              ;;IF ZERO GET OUT
11568  046510  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
11569  046514  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
11570  046516  122716  000067                  CMPB    #'7,(SP)
11571  046522  002423                          BLT     4$
11572  046524  006301                          ASL     R1              ;;*2
11573  046526  006102                          ROL     R2
11574  046530  006301                          ASL     R1              ;;*4
11575  046532  006102                          ROL     R2
11576  046534  006301                          ASL     R1              ;;*8
11577  046536  006102                          ROL     R2
11578  046540  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
11579  046544  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
11580  046546  000756                          BR      2$              ;;LOOP
11581  046550  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
11582  046552  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
11583  046556  010237  046610                  MOV     R2,$HIOCT
11584  046562  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11585  046564  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11586  046566  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11587  046570  000002                          RTI                     ;;RETURN
11588  046572  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
11589  046574  105010                          CLRB    (R0)            ;;SET A TERMINATOR
11590  046576  104400                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
11591  046600  000000                  5$:     .WORD   0
11592  046602  104400  001222                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
11593  046606  000730                          BR      1$              ;;TRY AGAIN
11594  046610  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                        .SBTTL  ERROR HANDLER ROUTINE

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

                                        $ERROR:
046612  104406                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR

046614                                  REGSAV:
046614  012737  177777  004734                  MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
046622                                  REGSA1:
046622  105237  001103                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
046626  001775                                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
046630  013777  001102  132304                  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
046636  032777  002000  132274                  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
046644  001402                                  BEQ     1$              ;;NO - SKIP
046646  104400  001216                          TYPE    ,$BELL          ;;RING BELL
046652  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
046656  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
046662  162737  000002  001116                  SUB     #2,$ERRPC
046670  117737  132222  001114                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
046676  032777  020000  132234                  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
046704  001004                                  BNE     20$             ;;SKIP TYPEOUTS
046706  004737  046762                          JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
046712  104400  001223                          TYPE    ,$CRLF
046716                                  20$:
046716  005777  132216          2$:     TST     @SWR            ;;HALT ON ERROR
046722  100002                                  BPL     3$              ;;SKIP IF CONTINUE
046724  000000                                  HALT                    ;;HALT ON ERROR!
046726  104406                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
046730  032777  001000  132202          3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
046736  001402                                  BEQ     4$              ;;BR IF NO
046740  013716  001110                          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
046744  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
046750  001402                                  BEQ     5$              ;;BR IF NONE
046752  013716  001214                          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
046756                                  5$:
046756  000002                                  RTI                     ;;RETURN
```

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

046760  000000                  PRITEM: 0                               ;PREVIOUS ITEM NO. LOCATION

046762  017746  132152          $ERRTYP: MOV     @SWR,-(SP)             ;GET SWITCH SETTING
046766  042716  177727                   BIC     #^C50,(SP)             ;KEEP ONLY SWITCH 8 AND 6
046772  022726  000100                   CMP     #SW06,(SP)+            ;IS 6 SET AND 8 RESET
046776  001001                           BNE     1$                     ;IF NOT BRANCH
047000  000402                           BR      2$                     ;BRANCH IF SW 6 IS SET AND 8 RESET
047002  000137  047722          1$:      JMP     @#TYPERR               ;JUMP IF SW 8 IS SET
                                                                        ;OR IF SW 8 IS RESET AND SW 6 IS RESET
047006                          2$:

047006  104400  047014                   TYPE    ,65$                   ;;TYPE ASCIZ STRING
047012  000406                           BR      64$                    ;;GET OVER THE ASCIZ
                                ;;65$:   .ASCIZ  <15><12>/RHWC =
047030                          64$:
047030  013746  002354                   MOV     @#WC,-(SP)             ;GET READY TO TYPE RHWC CONTENTS
047034  104401                           TYPOC

047036  104400  047044                   TYPE    ,67$                   ;;TYPE ASCIZ STRING
047042  000406                           BR      66$                    ;;GET OVER THE ASCIZ
                                ;;67$:   .ASCIZ  <15><12>/RHBA =
047060                          66$:
047060  013746  002356                   MOV     @#BA,-(SP)             ;GET READY TO TYPE RHBA CONTENTS
047064  104401                           TYPOC

047066  104400  047074                   TYPE    ,69$                   ;;TYPE ASCIZ STRING
047072  000406                           BR      68$                    ;;GET OVER THE ASCIZ
                                ;;69$:   .ASCIZ  <15><12>/RHCS2 =
047110                          68$:
047110  013746  002360                   MOV     @#CS2,-(SP)            ;GET READY TO TYPE RHCS2 CONTENTS
047114  104401                           TYPOC

047116  104400  047124                   TYPE    ,71$                   ;;TYPE ASCIZ STRING
047122  000406                           BR      70$                    ;;GET OVER THE ASCIZ
                                ;;71$:   .ASCIZ  <15><12>/RHCS1 =
```

```
11697  047140                                  70$:
11698  047140  013746  002362                          MOV     @#CS1,-(SP)     ;GET READY TO TYPE RHCS1 CONTENTS
11699  047144  104401                                  TYPOC
11700
11701
11702  047146  104400  047154                          TYPE    ,73$            ;;TYPE ASCIZ STRING
11703  047152  000406                                  BR      72$             ;;GET OVER THE ASCIZ
11704                                          ;;73$:  .ASCIZ  <15><12>/RHDS1 =  /
11705  047170                                  72$:
11706  047170  013746  002404                          MOV     @#DS1,-(SP)     ;GET READY TO TYPE RHDS1 CONTENTS
11707  047174  104401                                  TYPOC
11708
11709
11710  047176  104400  047204                          TYPE    ,75$            ;;TYPE ASCIZ STRING
11711  047202  000406                                  BR      74$             ;;GET OVER THE ASCIZ
11712                                          ;;75$:  .ASCIZ  <15><12>/RHER1 =  /
11713  047220                                  74$:
11714  047220  013746  002364                          MOV     @#ER1,-(SP)     ;GET READY TO TYPE RHER1 CONTENTS
11715  047224  104401                                  TYPOC
11716
11717
11718  047226  104400  047234                          TYPE    ,77$            ;;TYPE ASCIZ STRING
11719  047232  000406                                  BR      76$             ;;GET OVER THE ASCIZ
11720                                          ;;77$:  .ASCIZ  <15><12>/RHER2 =  /
11721  047250                                  76$:
11722  047250  013746  002370                          MOV     @#ER2,-(SP)     ;GET READY TO TYPE RHER2 CONTENTS
11723  047254  104401                                  TYPOC
11724
11725
11726  047256  104400  047264                          TYPE    ,79$            ;;TYPE ASCIZ STRING
11727  047262  000406                                  BR      78$             ;;GET OVER THE ASCIZ
11728                                          ;;79$:  .ASCIZ  <15><12>/RHER3 =  /
11729  047300                                  78$:
11730  047300  013746  002376                          MOV     @#ER3,-(SP)     ;GET READY TO TYPE RHER3 CONTENTS
11731  047304  104401                                  TYPOC
11732
11733
11734  047306  104400  047314                          TYPE    ,81$            ;;TYPE ASCIZ STRING
11735  047312  000406                                  BR      80$             ;;GET OVER THE ASCIZ
11736                                          ;;81$:  .ASCIZ  <15><12>/RHDST =  /
11737  047330                                  80$:
11738  047330  013746  002366                          MOV     @#DST,-(SP)     ;GET READY TO TYPE RHDST CONTENTS
11739  047334  104401                                  TYPOC
11740
11741
11742  047336  104400  047344                          TYPE    ,83$            ;;TYPE ASCIZ STRING
11743  047342  000406                                  BR      82$             ;;GET OVER THE ASCIZ
11744                                          ;;83$:  .ASCIZ  <15><12>/RHCA =  /
11745  047360                                  82$:
11746  047360  013746  002374                          MOV     @#CA,-(SP)      ;GET READY TO TYPE RHCA CONTENTS
11747  047364  104401                                  TYPOC
11748
11749
11750  047366  104400  047374                          TYPE    ,85$            ;;TYPE ASCIZ STRING
```

```
011751  047372  000406                          BR      94$             ;;GET OVER THE ASCIZ
011752                          ;;95$:  .ASCIZ  <15><12>/RHAS = /
011753  047410                  94$:
011754  047410  013746  002400                  MOV     @#AS,-(SP)      ;GET READY TO TYPE RHAS CONTENTS
011755  047414  104401                          TYPOC
011756
011757
011758  047416  104400  047424                  TYPE    ,97$            ;;TYPE ASCIZ STRING
011759  047422  000406                          BR      86$             ;;GET OVER THE ASCIZ
011760                          ;;97$:  .ASCIZ  <15><12>/RHOF = /
011761  047440                  86$:
011762  047440  013746  002372                  MOV     @#OF,-(SP)      ;GET READY TO TYPE RHOF CONTENTS
011763  047444  104401                          TYPOC
011764
011765
011766  047446  104400  047454                  TYPE    ,89$            ;;TYPE ASCIZ STRING
011767  047452  000406                          BR      88$             ;;GET OVER THE ASCIZ
011768                          ;;89$:  .ASCIZ  <15><12>/RHMR = /
011769  047470                  88$:
011770  047470  013746  002402                  MOV     @#MR,-(SP)      ;GET READY TO TYPE RHMR CONTENTS
011771  047474  104401                          TYPOC
011772
011773
011774  047476  104400  047504                  TYPE    ,91$            ;;TYPE ASCIZ STRING
011775  047502  000406                          BR      90$             ;;GET OVER THE ASCIZ
011776                          ;;91$:  .ASCIZ  <15><12>/RHLA = /
011777  047520                  90$:
011778  047520  013746  002420                  MOV     @#LA,-(SP)      ;GET READY TO TYPE RHLA CONTENTS
011779  047524  104401                          TYPOC
011780
011781
011782  047526  104400  047534                  TYPE    ,93$            ;;TYPE ASCIZ STRING
011783  047532  000406                          BR      92$             ;;GET OVER THE ASCIZ
011784                          ;;93$:  .ASCIZ  <15><12>/RHCC = /
011785  047550                  92$:
011786  047550  013746  002416                  MOV     @#CC,-(SP)      ;GET READY TO TYPE RHCC CONTENTS
011787  047554  104401                          TYPOC
011788
011789
011790  047556  104400  047564                  TYPE    ,95$            ;;TYPE ASCIZ STRING
011791  047562  000406                          BR      94$             ;;GET OVER THE ASCIZ
011792                          ;;95$:  .ASCIZ  <15><12>/RHEC1 = /
011793  047600                  94$:
011794  047600  013746  002412                  MOV     @#EC1,-(SP)     ;GET READY TO TYPE RHEC1 CONTENTS
011795  047604  104401                          TYPOC
011796
011797
011798  047606  104400  047614                  TYPE    ,97$            ;;TYPE ASCIZ STRING
011799  047612  000406                          BR      96$             ;;GET OVER THE ASCIZ
011800                          ;;97$:  .ASCIZ  <15><12>/RHEC2 = /
011801  047630                  96$:
011802  047630  013746  002414                  MOV     @#EC2,-(SP)     ;GET READY TO TYPE RHEC2 CONTENTS
011803  047634  104401                          TYPOC
011804
```

```
11905
11906  047636  104400  047644          TYPE    @99$            ;;TYPE ASCIZ STRING
11907  047642  000406                  BR      99$             ;;GET OVER THE ASCIZ
11908                          ;;99$:   .ASCIZ  <15><12>/RHDT = /
11909  047660                  99$:
11910  047660  013746  002406          MOV     @#DT,-(SP)      ;GET READY TO TYPE RHDT CONTENTS
11911  047664  104401                  TYPOC
11912
11913
11914  047666  104400  047674          TYPE    @101$           ;;TYPE ASCIZ STRING
11915  047672  000406                  BR      100$            ;;GET OVER THE ASCIZ
11916                          ;;101$:  .ASCIZ  <15><12>/RHSN = /
11917  047710                  100$:
11918  047710  013746  002410          MOV     @#SN,-(SP)      ;GET READY TO TYPE RHSN CONTENTS
11919  047714  104401                  TYPOC
11920
11921  047716  005037  046760          CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
11922  047722                  TYPERR:
11923  047722  104400  001223          TYPE    $CRLF           ;"CARRIAGE RETURN" & "LINE FEED"
11924  047726  010046                  MOV     R0,-(SP)        ;SAVE R0
11925  047730  005000                  CLR     R0              ;PICKUP THE ITEM INDEX
11926  047732  153700  001114          BISB    @#$ITEMB,R0
11927  047736  001004                  BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
11928                                                          ;TYPE THE PC OF THE ERROR
11929  047740  013746  001116          MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
11930                                                          ;ERROR ADDRESS
11931  047744  104401                  TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
11932  047746  000454                  BR      10$             ;GET OUT
11933  047750  005300          1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
11934  047752  006300                  ASL     R0              ;       WORK FOR THE ERROR TABLE
11935  047754  006300                  ASL     R0
11936  047756  006300                  ASL     R0
11937  047760  062700  001226          ADD     #$ERRTB,R0      ;FORM TABLE POINTER
11938  047764  020037  046760          CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
11939  047770  001002                  BNE     13$             ;BRANCH IF NOT
11940  047772  022020                  CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
11941  047774  000420                  BR      5$
11942  047776  010037  046760  13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
11943  050002  012037  050012          MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
11944  050006  001404                  BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
11945  050010  104400                  TYPE                    ;TYPE THE "ERROR MESSAGE"
11946  050012  000000          2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
11947  050014  104400  001223          TYPE    $CRLF           ;"CARRIAGE RETURN" & "LINE FEED"
11948  050020  012037  050030  3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
11949  050024  001404                  BEQ     5$              ;SKIP TYPEOUT IF 0
11950  050026  104400                  TYPE                    ;TYPE THE "DATA HEADER"
11951  050030  000000          4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
11952  050032  104400  001223          TYPE    $CRLF           ;"CARRIAGE RETURN" & "LINE FEED"
11953  050036  010146          5$:     MOV     R1,-(SP)        ;SAVE R1
11954  050040  012001                  MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
11955  050042  001415                  BEQ     9$              ;BR IF NO DATA TO BE TYPED
11956  050044  012000                  MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
11957  050046  105720          6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
11958  050050  001003                  BNE     7$              ;BR IF DECIMAL
```

```
 11859  050052  013146                          MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
 11860  050054  104401                          TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 11861  050056  000402                          BR      8$
 11862  050060                          7$:
 11863  050060  013146                          MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
 11864  050062  104404                          TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
 11865  050064  005711                  8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
 11866  050066  001403                          BEQ     9$              ;BR IF NO
 11867  050070  104400  050104                  TYPE    ,11$            ;TYPE TWO(2) SPACES
 11868  050074  000764                          BR      6$              ;LOOP
 11869
 11870  050076  012601                  9$:     MOV     (SP)+,R1        ;RESTORE R1
 11871  050100  012600                  10$:    MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
 11872  050102  000207                          RTS     PC              ;RETURN
 11873  050104  020040      040         11$:    .ASCIZ  /  /            ;TWO(2) SPACES
 11874  050110                                  .EVEN
 11875
```

```
                                        .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

011902  050110  012646  000000          $TYPOS: MOV     @(SP).-(SP)     ;;PICKUP THE MODE
011903  050114  116637  000001  050333          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
011904  050122  112637  050335                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
011905  050126  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
011906  050132  000406                          BR      $TYPON
011907  050134  112737  000001  050333  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
011908  050142  112737  000006  050335          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
011909  050150  112737  000005  050332  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
011910  050156  010346                          MOV     R3,-(SP)        ;;SAVE R3
011911  050160  010446                          MOV     R4,-(SP)        ;;SAVE R4
011912  050162  010546                          MOV     R5,-(SP)        ;;SAVE R5
011913  050164  113704  050335                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
011914  050170  005404                          NEG     R4
011915  050172  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
011916  050176  110437  050334                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
011917  050202  113704  050333                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
011918  050206  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
011919  050212  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
011920  050214  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
011921  050216  000404                          BR      3$              ;;GO DO MSB
011922  050220  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
011923  050222  006105                          ROL     R5
011924  050224  006105                          ROL     R5
011925  050226  010503                          MOV     R5,R3
011926  050230  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
011927  050232  105337  050334                  DECB    $OMODE          ;;TYPE THIS DIGIT?
011928  050236  100016                          BPL     7$              ;;BR IF NO
011929  050240  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
```

```
11930  050244  001002                        BNE     4$              ;;TEST FOR 0
11931  050246  005704                        TST     R4              ;;SUPPRESS THIS 0?
11932  050250  001403                        BEQ     5$              ;;BR IF YES
11933  050252  005204              4$:       INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11934  050254  052703  000060              BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11935  050260  052703  000040              5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREAD.
11936  050264  110337  050330                      MOVB    R3,8$           ;;SAVE FOR TYPING
11937  050270  104400  050330                      TYPE    ,8$             ;;GO TYPE THIS DIGIT
11938  050274  105337  050332              7$:     DECB    $OCNT           ;;COUNT BY 1
11939  050300  003347                        BGT     2$              ;;BR IF MORE TO DO
11940  050302  002402                        BLT     6$              ;;BR IF DONE
11941  050304  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11942  050306  000744                        BR      2$              ;;GO DO THE LAST DIGIT
11943  050310  012605              6$:       MOV     (SP)+,R5        ;;RESTORE R5
11944  050312  012604                        MOV     (SP)+,R4        ;;RESTORE R4
11945  050314  012603                        MOV     (SP)+,R3        ;;RESTORE R3
11946  050316  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11947  050324  012616                        MOV     (SP)+,(SP)
11948  050326  000002                        RTI                     ;;RETURN
11949  050330  000              8$:       .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11950  050331  000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11951  050332  003              $OCNT:    .BYTE   3               ;;OCTAL DIGIT COUNTER
11952  050333  000              $OFILL:   .BYTE   0               ;;ZERO FILL SWITCH
11953  050334  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.

050336  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
050340  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
050344  005740                          TST     -(R0)           ;;BACKUP BY 2
050346  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
050350  006300                          ASL     R0              ;;POSITION FOR INDEXING
050352  016000  050360                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
050356  000200                          RTS     R0              ;;GO TO ROUTINE

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
050360                          $TRPAD:
050360  045112                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
050362  050134                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
050364  050110                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
050366  050150                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
050370  044666                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)

050372  045730                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING

050374  045640                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
050376  046202                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
050400  046272                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
050402  046452                          $RDOCT  ;;CALL=RDOCT    TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
050404  042066                          WAIT.T  ;;CALL=WAT      TRAP+12(104412) DONT ADD ABOVE THIS TRAP
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
050406  012737  050552  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
050414  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
050422  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
050424  010146                          MOV     R1,-(SP)         ;;PUSH R1 ON STACK
050426  010246                          MOV     R2,-(SP)         ;;PUSH R2 ON STACK
050430  010346                          MOV     R3,-(SP)         ;;PUSH R3 ON STACK
050432  010446                          MOV     R4,-(SP)         ;;PUSH R4 ON STACK
050434  010546                          MOV     R5,-(SP)         ;;PUSH R5 ON STACK
050436  017746  130476                  MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
050442  010637  050556                  MOV     SP,$SAVR6        ;;SAVE SP
050446  012737  050460  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
050454  000000                          HALT
050456  000776                          BR      .-2              ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
050460  012737  050552  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
050466  013706  050556                  MOV     $SAVR6,SP        ;;GET SP
050472  005037  050556                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
050476  005237  050556          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
050502  001375                          BNE     1$               ;;OF  WORD
050504  012677  130430                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
050510  012605                          MOV     (SP)+,R5         ;;POP STACK INTO R5
050512  012604                          MOV     (SP)+,R4         ;;POP STACK INTO R4
050514  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
050516  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
050520  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
050522  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
050524  012737  050406  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
050532  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
050540  104400                          TYPE                     ;REPORT THE POWER FAILURE
050542  050560          $PWRMG: .WORD   $POWER                   ;;POWER FAIL MESSAGE POINTER
050544  012716                  MOV     (PC)+,(SP)               ;;RESTART AT BEGIN
050546  005012          $PWRAD: .WORD   BEGIN                    ;;RESTART ADDRESS
050550  000002                  RTI
050552  000000          $ILLUP: HALT                             ;;THE POWER UP SEQUENCE WAS STARTED
050554  000776                  BR      .-2                      ;; BEFORE THE POWER DOWN WAS COMPLETE
050556  000000          $SAVR6: 0                                ;;PUT THE SP HERE
050560  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
050566  000122
                                        .EVEN
```

```
12037                                   ;************************************
12038                                   ;*
12039                                   ;*ERROR AND MESSAGE TABLE CONDIMENTS
12040                                   ;*
12041                                   ;************************************
12042
12043
12044
12045
12046  050570  050122  032060  042040  EM1:    .ASCIZ  /RP04 DID NOT INTERRUPT/
12047  050576  042111  047040  052117
12048  050604  044440  052116  051105
12049  050612  052522  052120     000
12050  050617     111  052116  051105  EM2:    .ASCIZ  /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
12051  050624  052522  052120  042440
12052  050632  040516  046102  020105
12053  050640  044502  020124  047504
12054  050646  047127  041040  052125
12055  050654  042440  050130  041505
12056  050662  042524  020104  044502
12057  050670  020124  044504  020104
12058  050676  047516  020124  042523
12059  050704  000124
12060  050706  050122  032060  042040  EM3:    .ASCIZ  /RP04 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
12061  050714  042111  047040  052117
12062  050722  044440  052116  051105
12063  050730  052522  052120  053440
12064  050736  042510  020116  054105
12065  050744  042520  052103  042105
12066  050752  041040  052111  042040
12067  050760  042111  051440  052105
12068  050766     000
12069  050767     105  050130  041505  EM4:    .ASCIZ  /EXPECTED BIT DID SET BUT TIME IS IN ERROR (TIME IN 10 MICROSEC, DECIMAL
12070  050774  042524  020104  044502
12071  051002  020124  044504  020104
12072  051010  042523  020124  052502
12073  051016  020124  044524  042515
12074  051024  044440  020123  047111
12075  051032  042440  051122  051117
12076  051040  024040  044524  042515
12077  051046  044440  020116  030061
12078  051054  046440  041511  047522
12079  051062  042523  026103  042040
12080  051070  041505  046511  046101
12081  051076  000051
12082  051100  044122  051501  042040  EM5:    .ASCIZ  /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
12083  051106  042517  020123  047516
12084  051114  020124  046103  040505
12085  051122  020122  054502  046440
12086  051130  053117  047111  020107
12087  051136  047111  040440  046114
12088  051144  047440  042516  000123
12089  051152  047514  042101  047111  EM6:    .ASCIZ  /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
12090  051160  020107  044122  051105
```

```
12091  051166  020061  047506  020122
12092  051174  046101  020114  047125
12093  051202  052111  020123  044504
12094  051210  020104  047516  020124
12095  051216  042523  020124  047101
12096  051224  020131  044122  051501
12097  051232  041040  052111  000123
12098  051240  050123  041505  043111  EM7:    .ASCIZ  /SPECIFIED REGISTER NON EXISTANT, SO ABORT PROGRAM/
12099  051246  042511  020104  042522
12100  051254  044507  052123  051105
12101  051262  047040  047117  042440
12102  051270  044530  052123  047101
12103  051276  026124  051440  020117
12104  051304  041101  051117  020124
12105  051312  051120  042517  040522
12106  051320  000115
12107
12108  051322  052123  050117  042520  EM10:   .ASCIZ  /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
12109  051330  020104  051104  053111
12110  051336  020105  040510  020123
12111  051344  047515  020114  044502
12112  051352  020124  047111  051040
12113  051360  042110  030523  051440
12114  051366  052105     000
12115  051371     127  052111  020110  EM11:   .ASCIZ  /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO. AND IR SET/
12116  051376  050123  047111  046104
12117  051404  020105  047520  042527
12118  051412  042522  020104  047504
12119  051420  047127  051040  041510
12120  051426  031123  051440  047510
12121  051434  046125  020104  047117
12122  051442  054514  044040  053101
12123  051450  020105  047125  052111
12124  051456  047040  027117  040440
12125  051464  042116  044440  020122
12126  051472  042523  000124
12127  051476  043101  042524  020122  EM12:   .ASCIZ  /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
12128  051504  050123  047111  046104
12129  051512  020105  047520  042527
12130  051520  042522  020104  050125
12131  051526  020054  047516  050040
12132  051534  041501  020113  041501
12133  051542  047113  020056  044122
12134  051550  051504  020061  044123
12135  051556  052517  042114  044040
12136  051564  053101  020105  047515
12137  051572  036514  026061  053040
12138  051600  036526  000060
12139  051604  044527  044124  051440  EM13:   .ASCIZ  /WITH SPINDLE POWERED UP, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, R
12140  051612  044520  042116  042514
12141  051620  050040  053517  051105
12142  051626  042105  052440  026120
12143  051634  047040  020117  047111
12144  051642  044524  046101  055111
```

```
12145  051650  026105  051040  041510
12146  051656  030523  051440  047510
12147  051664  046125  020104  040510
12148  051672  042526  043440  036517
12149  051700  026060  042040  040526
12150  051706  030475  020054  042122
12151  051714  036531  026061  044440
12152  051722  036505  000060
12153  051726  043101  042524  020122  EM14:   .ASCIZ  /AFTER SPINDLE POWERED UP, RHCC SHOULD BE=0/
12154  051734  050123  047111  046104
12155  051742  020105  047520  042527
12156  051750  042522  020104  050125
12157  051756  020054  044122  041503
12158  051764  051440  047510  046125
12159  051772  020104  042502  030075
12160  052000     000
12161  052001     120  041501  020113  EM15:   .ASCII  <PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
12162  052006  041501  047113  053517
12163  052014  042514  043504  020105
12164  052022  047503  046515  047101
12165  052030  020104  040503  051525
12166  052036  042105  040440  020116
12167  052044  051105  047522  006522
12168  052052     012
12169  052053     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12170  052060  040504  040524  044440
12171  052066  020123  042502  047506
12172  052074  042522  041440  046517
12173  052102  040515  042116  020054
12174  052110  042522  020103  040504
12175  052116  040524  044440  020123
12176  052124  043101  042524  020122
12177  052132  047503  046515  047101
12178  052140  000104
12179  052142  047516  047455  020120  EM16:   .ASCII  /NO-OP COMMAND CAUSED AN ERROR/<15><12>
12180  052150  047503  046515  047101
12181  052156  020104  040503  051525
12182  052164  042105  040440  020116
12183  052172  051105  047522  006522
12184  052200     012
12185  052201     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12186  052206  040504  040524  044440
12187  052214  020123  042502  047506
12188  052222  042522  041440  046517
12189  052230  040515  042116  020054
12190  052236  042522  020103  040504
12191  052244  040524  044440  020123
12192  052252  043101  042524  020122
12193  052260  047503  046515  047101
12194  052266  000104
12195  052270  051104  053111  020105  EM17:   .ASCII  /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
12196  052276  046103  040505  020122
12197  052304  047503  046515  047101
12198  052312  020104  040503  051525
```

```
 12198  052320  042105  040440  020116
 12199  052326  051105  047522  006522
 12200  052334     012
 12201  052335     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
 12202  052343  040504  040524  043440
 12203  052350  053111  051505  051440
 12204  052356  047510  046125  020104
 12205  052364  042503  020054  042522
 12206  052372  020103  040504  040524
 12207  052400  043440  053111  051505
 12208  052406  040440  052106  051105
 12209  052414  041440  046517  040515
 12210  052422  042116     000
 12211
 12212
 12213  052425     122  040505  026504  EM20:   .ASCII  /READ-IN COMMAND CAUSED AN ERROR/<15><12>
 12214  052432  047111  041440  046517
 12215  052440  040515  042116  041440
 12216  052446  052501  042523  020104
 12217  052454  047101  042440  051122
 12218  052462  051117  005015
 12219  052466  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
 12220  052474  052101  020101  044507
 12221  052502  042526  020123  044123
 12222  052510  052517  042114  041040
 12223  052516  026105  051040  041505
 12224  052524  042040  052101  020101
 12225  052532  044507  042526  020123
 12226  052540  042522  027107  041440
 12227  052546  047117  027124  040440
 12228  052554  052106  051105  041440
 12229  052562  046517  040515  042116
 12230  052570     000
 12231  052571     122  041510  030523  EM21:   .ASCIZ  /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
 12232  052576  041440  047117  042524
 12233  052604  052116  020123  052504
 12234  052612  044522  043516  041440
 12235  052620  046517  040515  042116
 12236  052626  053440  051501  044440
 12237  052634  020116  051105  047522
 12238  052642  000122
 12239  052644  044122  051504  020061  EM22:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
 12240  052652  047503  052116  047105
 12241  052660  051524  042040  051125
 12242  052666  047111  020107  047503
 12243  052674  046515  047101  020104
 12244  052702  040527  020123  047111
 12245  052710  042440  051122  051117
 12246  052716     000
 12247  052717     125  046116  040517  EM23:   .ASCII  /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
 12248  052724  020104  047503  046515
 12249  052732  047101  020104  040503
 12250  052740  051525  042105  040440
 12251  052746  020116  051105  047522
 12252  052754  006522     012
```

```
12253  052757     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12254  052764  040504  040524  043440
12255  052772  053111  051505  051440
12256  053000  047510  046125  020104
12257  053006  042502  020054  042522
12258  053014  020103  040504  040524
12259  053022  043440  053111  051505
12260  053030  051040  043505  020056
12261  053036  047503  052116  020056
12262  053044  043101  042524  020122
12263  053052  047503  046515  047101
12264  053060  000104
12265  053062  043117  051506  052105  EM24:   .ASCII  /OFFSET COMMAND CAUSED AN ERROR/<15><12>
12266  053070  041440  046517  040515
12267  053076  042116  041440  052501
12268  053104  042523  020104  047101
12269  053112  042440  051122  051117
12270  053120  005015
12271  053122  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12272  053130  052101  020101  044507
12273  053136  042526  020123  044123
12274  053144  052517  042114  041040
12275  053152  026105  051040  041505
12276  053160  042040  052101  020101
12277  053166  044507  042526  020123
12278  053174  042522  027107  041440
12279  053202  047117  027124  040440
12280  053210  052106  051105  041440
12281  053216  046517  040515  042116
12282  053224     000
12283  053225     122  052105  051125  EM25:   .ASCII  /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
12284  053232  020116  047524  041440
12285  053240  047105  042524  020122
12286  053246  044514  042516  041440
12287  053254  046517  040515  042116
12288  053262  041440  052501  042523
12289  053270  020104  047101  042440
12290  053276  051122  051117  005015
12291  053304  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12292  053312  052101  020101  044507
12293  053320  042526  020123  044123
12294  053326  052517  042114  041040
12295  053334  026105  051040  041505
12296  053342  042040  052101  020101
12297  053350  044507  042526  020123
12298  053356  042522  027107  041440
12299  053364  047117  027124  040440
12300  053372  052106  051105  041440
12301  053400  046517  040515  042116
12302  053406     000
12303  053407     065  030060  047440  EM26:   .ASCIZ  /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR
12304  053414  043106  042523  020124
12305  053422  047503  046515  047101
12306  053430  051504  047440  042516
```

```
12337  053436  040440  052106  051105
12338  053444  052040  042510  047440
12339  053452  044124  051105  041440
12310  053460  052501  042523  020104
12311  053466  047101  042440  051122
12312  053474  051117     000
12313  053477     127  044522  042524  EM27:   .ASCII   WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE<15><12>
12314  053504  044040  040505  042504
12315  053512  020122  047101  020104
12316  053520  040504  040524  041440
12317  053526  052501  042523  020104
12318  053534  046511  051120  050117
12319  053542  051105  051040  043505
12320  053550  051511  042524  020122
12331  053556  044103  047101  042507
12332  053564  005015
12323  053566  047507  042117  042040          .ASCII  <GOOD DATA GIVES WHAT SHOULD BE THERE><15><12>
12324  053574  052101  020101  044507
12325  053602  042526  020123  044127
12326  053610  052101  051440  047510
12327  053616  046125  020104  042502
12328  053624  052040  042510  042522
12329  053632  005015
12330  053634  042522  042503  053111          .ASCII  RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND<
12331  053642  042105  042040  052101
12332  053650  020101  044507  042526
12333  053656  020123  044127  052101
12334  053664  053440  051501  052040
12335  053672  042510  042522  040440
12336  053700  052106  051105  041440
12337  053706  046517  040515  042116
12338  053714     000
12339
12340  053715     127  044522  042524  EM30:   .ASCIZ  <WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER><
12341  053722  044040  040505  042504
12342  053730  020122  047101  020104
12343  053736  040504  040524  041440
12344  053744  040510  043516  042105
12345  053752  053440  044522  042524
12346  053760  043040  047522  020115
12347  053766  052502  043106  051105
12348  053774     000
12349  053775     122  040505  020104  EM31:   .ASCII  <READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE><15><12>
12350  054002  042510  042101  051105
12351  054010  040440  042116  042040
12352  054016  052101  020101  040503
12353  054024  051525  042105  044440
12354  054032  050115  047522  042520
12355  054040  020122  042522  044507
12356  054046  052123  051105  041440
12357  054054  040510  043516  006505
12358  054062     012
12359  054063     107  047517  020104          .ASCII  <GOOD DATA GIVES WHAT SHOULD BE THERE><15><12>
12360  054070  040504  040524  043440
```

```
12361  054076  053111  051505  053440
12362  054104  040510  020124  044123
12363  054112  052517  042114  041040
12364  054120  020105  044124  051105
12365  054126  006505     012
12366  054131     122  041505  044505          .ASCIZ   RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
12367  054136  042526  020104  040504
12368  054144  040524  043440  053111
12369  054152  051505  053440  040510
12370  054160  020124  040527  020123
12371  054166  044124  051105  020105
12372  054174  043101  042524  020122
12373  054202  047503  046515  047101
12374  054210  000104
12375  054212  051127  052111  020105  EM32:   .ASCIZ   WRITE HEADER DATA, FOLLOWED BY READ HEADER AND DATA, CAUSED DATA ERROR/
12376  054220  042510  042101  051105
12377  054226  042040  052101  026101
12378  054234  043040  046117  047514
12379  054242  042527  020104  054502
12380  054250  051040  040505  020104
12381  054256  042510  042101  051105
12382  054264  040440  042116  042040
12383  054272  052101  026101  041440
12384  054300  052501  042523  020104
12385  054306  040504  040524  042440
12386  054314  051122  051117     000
12387  054321     122  040505  020104  EM33:   .ASCII   READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
12388  054326  040504  040524  041440
12389  054334  052501  042523  020104
12390  054342  046511  051120  050117
12391  054350  051105  051040  043505
12392  054356  051511  042524  020122
12393  054364  044103  047101  042507
12394  054372  005015
12395  054374  047507  042117  042040          .ASCII   GOOD DATA GIVES WHAT SHOULD BE THERE<15><12>
12396  054402  052101  020101  044507
12397  054410  042526  020123  044127
12398  054416  052101  051440  047510
12399  054424  046125  020104  042502
12400  054432  052040  042510  042522
12401  054440  005015
12402  054442  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
12403  054450  042105  042040  052101
12404  054456  020101  044507  042526
12405  054464  020123  044127  052101
12406  054472  053440  051501  052040
12407  054500  042510  042522  040440
12408  054506  052106  051105  041440
12409  054514  046517  040515  042116
12410  054522     000
12411  054523     122  040505  020104  EM34:   .ASCIZ   READ DATA INCORRECT
12412  054530  042504  040524  044440
12413  054536  041516  051117  042522
12414  054544  052103     000
```

```
12415  054547    127  044522  042524  EM35:   .ASCII   WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE<15><12>
12416  054554 042040  052101  020101
12417  054562 047503  046515  047101
12418  054570 020104  040503  051525
12419  054576 042105  044440  050115
12420  054604 047522  042520  020122
12421  054612 042522  044507  052123
12422  054620 051105  041440  040510
12423  054626 043516  006505    012
12424  054633    107  047517  020104          .ASCII   GOOD DATA GIVES WHAT SHOULD BE THERE<15><12>
12425  054640 040504  040524  043440
12426  054646 053111  051505  053440
12427  054654 040510  020124  044123
12428  054662 052517  042114  041040
12429  054670 020105  044124  051105
12430  054676 006505    012
12431  054701    122  041505  044505          .ASCIZ   /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
12432  054706 042526  020104  040504
12433  054714 040524  043440  053111
12434  054722 051505  051040  043505
12435  054730 051511  042524  020122
12436  054736 047503  052116  047105
12437  054744 051524  040440  052106
12438  054752 051105  041440  046517
12439  054760 040515  042116    000
12440  054765    127  044522  042524  EM36:   .ASCIZ   WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
12441  054772 042040  052101  020101
12442  055000 047503  046515  047101
12443  055006 020104  044103  047101
12444  055014 042507  020104  051127
12445  055022 052111  020105  051106
12446  055030 046517  041040  043125
12447  055036 042506  000122
12448  055042 042523  045505  041440  EM37:   .ASCII   /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE<15><12>
12449  055050 046517  040515  042116
12450  055056 041440  052501  042523
12451  055064 020104  046511  051120
12452  055072 050117  051105  051040
12453  055100 043505  051511  042524
12454  055106 020122  044103  047101
12455  055114 042507  005015
12456  055120 047507  042117  042040          .ASCII   /GOOD DATA GIVES WHAT SHOULD BE THERE<15><12>
12457  055126 052101  020101  044507
12458  055134 042526  020123  044127
12459  055142 052101  051440  047510
12460  055150 046125  020104  042502
12461  055156 052040  042510  042522
12462  055164 005015
12463  055166 042522  042503  053111          .ASCIZ   /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND
12464  055174 042105  042040  052101
12465  055202 020101  044507  042526
12466  055210 020123  042522  044507
12467  055216 052123  051105  041440
12468  055224 047117  042524  052116
```

```
12469  055232  020123  043101  042524
12470  055240  020122  042523  045505
12471  055246  041440  046517  040515
12472  055254  042116     000
12473
12474  055257     127  044522  042524  EM40:   .ASCII   WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
12475  055264  041440  042510  045503
12476  055272  041440  052501  042523
12477  055300  020104  046511  051120
12478  055306  050117  051105  051040
12479  055314  043505  051511  042524
12480  055322  020122  044103  047101
12481  055330  042507  005015
12482  055334  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12483  055342  052101  020101  044507
12484  055350  042526  020123  044127
12485  055356  052101  051440  047510
12486  055364  046125  020104  042502
12487  055372  052040  042510  042522
12488  055400  005015
12489  055402  042522  042503  053111          .ASCIZ   RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
12490  055410  042105  042040  052101
12491  055416  020101  044507  042526
12492  055424  020123  042522  044507
12493  055432  052123  051105  041440
12494  055440  047117  042524  052116
12495  055446  020123  043101  042524
12496  055454  020122  047503  046515
12497  055462  047101  000104
12498  055466  047514  045503  047111  EM41:   .ASCII  /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
12499  055474  020107  052517  020124
12500  055502  051127  052111  020105
12501  055510  054502  053440  044522
12502  055516  042524  046040  041517
12503  055524  020113  052502  052124
12504  055532  047117  041440  052501
12505  055540  042523  020104  046511
12506  055546  051120  050117  051105
12507  055554  051040  043505  051511
12508  055562  042524  020122  044103
12509  055570  047101  042507  005015
12510  055576  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12511  055604  052101  020101  044507
12512  055612  042526  020123  044127
12513  055620  052101  051440  047510
12514  055626  046125  020104  042502
12515  055634  052040  042510  042522
12516  055642  005015
12517  055644  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
12518  055652  042105  042040  052101
12519  055660  020101  044507  042526
12520  055666  020123  042522  044507
12521  055674  052123  051105  041440
12522  055702  047117  042524  052116
```

```
12523  055710  020123  043101  042524
12524  055716  020122  051127  052111
12525  055724  051505  053440  051105
12526  055732  020105  047514  045503
12527  055740  042105  047440  052125
12528  055746     000
12529  055747     101  052124  046505  EM42:   .ASCII  /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
12530  055754  052120  047111  020107
12531  055762  047524  053440  044522
12532  055770  042524  053440  052111
12533  055776  020110  051127  052111
12534  056004  051505  046040  041517
12535  056012  042513  020104  052517
12536  056020  020124  040503  051525
12537  056026  042105  044440  050115
12538  056034  047522  042520  020122
12539  056042  042522  044507  052123
12540  056050  051105  041440  040510
12541  056056  043516  006505     012
12542  056063     107  047517  020104          .ASCII   GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12543  056070  040504  040524  043440
12544  056076  053111  051505  053440
12545  056104  040510  020124  044123
12546  056112  052517  042114  041040
12547  056120  020105  044124  051105
12548  056126  006505     012
12549  056131     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
12550  056136  042526  020104  040504
12551  056144  040524  043440  053111
12552  056152  051505  051040  043505
12553  056160  051511  042524  020122
12554  056166  047503  052116  047105
12555  056174  051524  040440  052106
12556  056202  051105  040440  052124
12557  056210  046505  052120  042105
12558  056216  053440  044522  042524
12559  056224     000
12560  056225     127  044522  044524  EM43:   .ASCII  /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA <15><12>
12561  056232  043516  053440  052111
12562  056240  020110  051127  052111
12563  056246  051505  046040  041517
12564  056254  042513  020104  052517
12565  056262  020124  044103  047101
12566  056270  042507  020104  044504
12567  056276  045523  042040  052101
12568  056304  006501     012
12569  056307     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE (WITH WRITE LOCKED OUT)<
12570  056314  040504  040524  043440
12571  056322  053111  051505  053440
12572  056330  040510  020124  040527
12573  056336  020123  047117  042040
12574  056344  051511  020113  042502
12575  056352  047506  042522  053440
12576  056360  044522  042524  024040
```

```
12577  056366  044527  044124  053440
12578  056374  044522  042524  046040
12579  056402  041517  042513  020104
12580  056410  052517  024524  005015
12581  056416  040527  020123  052101          .ASCII  /WAS ATTEMPTED/<15><12>
12582  056424  042524  050115  042524
12583  056432  006504     012
12584  056435     122  041505  044505          .ASCII  /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE <15> 12>
12585  056442  042526  020104  040504
12586  056450  040524  043440  053111
12587  056456  051505  053440  040510
12588  056464  020124  040527  020123
12589  056472  042522  042101  041040
12590  056500  041501  020113  043101
12591  056506  042524  020122  051127
12592  056514  052111  006505     012
12593  056521     050  044527  044124          .ASCIZ  /(WITH WRITE LOCKED OUT) WAS ATTEMPTED/
12594  056526  053440  044522  042524
12595  056534  046040  041517  042513
12596  056542  020104  052517  024524
12597  056550  053440  051501  040440
12598  056556  052124  046505  052120
12599  056564  042105     000
12600  056567     105  040516  046102  EM44:   .ASCII  /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
12601  056574  047111  020107  051127
12602  056602  052111  051505  041040
12603  056610  020131  051127  052111
12604  056616  020105  047514  045503
12605  056624  041040  052125  047524
12606  056632  020116  040503  051525
12607  056640  042105  044440  050115
12608  056646  047522  042520  020122
12609  056654  042522  044507  052123
12610  056662  051105  041440  040510
12611  056670  043516  006505     012
12612  056675     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12613  056702  040504  040524  043440
12614  056710  053111  051505  053440
12615  056716  040510  020124  044123
12616  056724  052517  042114  041040
12617  056732  020105  044124  051105
12618  056740  006505     012
12619  056743     122  041505  044505          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
12620  056750  042526  020104  040504
12621  056756  040524  043440  053111
12622  056764  051505  051040  043505
12623  056772  051511  042524  020122
12624  057000  047503  052116  047105
12625  057006  051524  040440  052106
12626  057014  051105  053440  044522
12627  057022  042524  046040  041517
12628  057030  020113  052502  052124
12629  057036  047117  005015
12630  057042  047105  041101  042514          .ASCIZ  /ENABLED WRITES/
```

```
12631  057050  020104  051127  052111
12632  057056  051505     000
12633  057061     124  040522  051516  EM45:   .ASCII   TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
12634  057066  042506  051122  047111
12635  057074  020107  047117  046040
12636  057102  051501  020124  046102
12637  057110  041517  020113  020055
12638  057116  054503  044514  042116
12639  057124  051105  032040  030061
12640  057132  020054  042523  052103
12641  057140  051117  031040  026061
12642  057146  052040  040522  045503
12643  057154  030440  006470     012
12644  057161     103  052501  042523          .ASCII   CAUSED IMPROPER REGISTER CHANGE/<15><12>
12645  057166  020104  046511  051120
12646  057174  050117  051105  051040
12647  057202  043505  051511  042524
12648  057210  020122  044103  047101
12649  057216  042507  005015
12650  057222  047507  042117  042040          .ASCII   GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12651  057230  052101  020101  044507
12652  057236  042526  020123  044127
12653  057244  052101  051440  047510
12654  057252  046125  020104  042502
12655  057260  052040  042513  042522
12656  057266  005015
12657  057270  042522  042503  053111          .ASCIZ   RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
12658  057276  042105  042040  052101
12659  057304  020101  044507  042526
12660  057312  020123  042522  044507
12661  057320  052123  051105  041440
12662  057326  047117  042524  052116
12663  057334  020123  043101  042524
12664  057342  020122  051124  047101
12665  057350  043123  051105     000
12666  057355     104  052101  020101  EM46:   .ASCII   /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
12667  057362  042522  042101  043040
12668  057370  047522  020115  040514
12669  057376  052123  041040  047514
12670  057404  045503  026440  041440
12671  057412  046131  047111  042504
12672  057420  020122  030464  026060
12673  057426  051440  041505  047524
12674  057434  020122  030462  020054
12675  057442  051124  041501  020113
12676  057450  034061  005015
12677  057454  051511  044440  020116          .ASCIZ   IS IN ERROR/
12678  057462  051105  047522  000122
12679  057470  051124  047101  043123  EM47:   .ASCII   /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER  <15><12>
12680  057476  051105  044522  043516
12681  057504  042040  052101  020101
12682  057512  051106  046517  047040
12683  057520  047117  054105  051511
12684  057526  040524  052116  051440
```

```
12685  057534  041505  047524  020122
12686  057542  040503  051525  042105
12687  057550  044440  050115  047522
12688  057556  042520  020122  005015
12689  057564  042522  044507  052123          .ASCII  /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12690  057572  051105  041440  040510
12691  057600  043516  026105  043440
12692  057606  047517  020104  040504
12693  057614  040524  043440  053111
12694  057622  051505  053440  040510
12695  057630  020124  044123  052517
12696  057636  042114  041040  020105
12697  057644  044124  051105  006505
12698  057652     012
12699  057653     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
12700  057660  042526  020104  040504
12701  057666  040524  043440  053111
12702  057674  051505  051040  043505
12703  057702  051511  042524  020122
12704  057710  047503  052116  047105
12705  057716  051524  040440  052106
12706  057724  051105  040440  052124
12707  057732  046505  052120  042105
12708  057740  052040  040522  051516
12709  057746  042506  000122
12710
12711  057752  051124  047101  043123  EM50:   .ASCII  /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
12712  057760  051105  044522  043516
12713  057766  043040  047522  020115
12714  057774  047516  042516  044530
12715  060002  052123  047101  020124
12716  060010  042523  052103  051117
12717  060016  041440  052501  042523
12718  060024  020104  040504  040524
12719  060032  042440  051122  051117
12720  060040  005015
12721  060042  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12722  060050  052101  020101  044507
12723  060056  042526  020123  044127
12724  060064  052101  051440  047510
12725  060072  046125  020104  042502
12726  060100  052040  042510  042522
12727  060106  005015
12728  060110  040502  020104  040504          .ASCIZ  /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
12729  060116  040524  043440  053111
12730  060124  051505  053440  040510
12731  060132  020124  040527  020123
12732  060140  047111  041040  043125
12733  060146  042506  020122  043101
12734  060154  042524  020122  051124
12735  060162  047101  043123  051105
12736  060170     000
12737  060171     107  053111  047111  EM51:   .ASCII  /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
12738  060176  020107  046111  042514
```

```
12739  060204  040507  020114  052506
12740  060212  041516  044524  047117
12741  060220  041440  052501  042523
12742  060226  020104  046511  051120
12743  060234  050117  051105  051040
12744  060242  043505  051511  042524
12745  060250  020122  044103  047101
12746  060256  042507  005015
12747  060262  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12748  060270  052101  020101  044507
12749  060276  042526  020123  044127
12750  060304  052101  051440  047510
12751  060312  046125  020104  042502
12752  060320  052040  042510  042522
12753  060326  005015
12754  060330  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
12755  060336  042105  042040  052101
12756  060344  020101  044507  042526
12757  060352  020123  042522  044507
12758  060360  052123  051105  041440
12759  060366  047117  042524  052116
12760  060374  020123  043101  042524
12761  060402  020122  046111  042514
12762  060410  040507  020114  052506
12763  060416  041516  044524  047117
12764  060424  044440  020123  044507
12765  060432  042526  000116
12766  060436  051127  052111  020105  EMS2:   .ASCII  /WRITE DATA COMMAND ON NONEXISTANT SECTOR CAUSED IMPROPER REG. CHANGE/<1
12767  060444  040504  040524  041440
12768  060452  046517  040515  042116
12769  060460  047440  020116  047516
12770  060466  042516  044530  052123
12771  060474  047101  020124  042523
12772  060502  052103  051117  041440
12773  060510  052501  042523  020104
12774  060516  046511  051120  050117
12775  060524  051105  051040  043505
12776  060532  020056  044103  047101
12777  060540  042507  005015
12778  060544  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12779  060552  052101  020101  044507
12780  060560  042526  020123  044127
12781  060566  052101  051440  047510
12782  060574  046125  020104  042502
12783  060602  052040  042510  042522
12784  060610  005015
12785  060612  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
12786  060620  042105  042040  052101
12787  060626  020101  044507  042526
12788  060634  020123  042522  044507
12789  060642  052123  051105  041440
12790  060650  047117  042524  052116
12791  060656  020123  043101  042524
12792  060664  020122  052101  042524
```

```
12793  060672  050115  042524  020104
12794  060700  051127  052111  020105
12795  060706  040504  040524     000
12796  060713     122  040505  020104  EM53:   .ASCIZ  /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
12797  060720  042510  042101  051105
12798  060726  040440  042116  042040
12799  060734  052101  020101  043101
12800  060742  042524  020122  020101
12801  060750  042523  051101  044103
12802  060756  041440  052501  042523
12803  060764  020104  040504  040524
12804  060772  042440  051122  051117
12805  061000     000
12806  061001     101  052124  046505  EM54:   .ASCII  /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
12807  061006  052120  047111  020107
12808  061014  047503  046515  047101
12809  061022  020104  044527  044124
12810  061030  044440  053116  046101
12811  061036  042111  040440  042104
12812  061044  042522  051523  041440
12813  061052  052501  042523  020104
12814  061060  046511  051120  050117
12815  061066  051105  051040  043505
12816  061074  051511  042524  020122
12817  061102  044103  047101  042507
12818  061110  005015
12819  061112  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12820  061120  052101  020101  044507
12821  061126  042526  020123  044127
12822  061134  052101  051440  047510
12823  061142  046125  020104  042502
12824  061150  052040  042510  042522
12825  061156  005015
12826  061160  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
12827  061166  042105  042040  052101
12828  061174  020101  044507  042526
12829  061202  020123  042522  044507
12830  061210  052123  051105  041440
12831  061216  047117  042524  052116
12832  061224  020123  043101  042524
12833  061232  020122  050117  051105
12834  061240  052101  047511  000116
12835  061246  051127  052111  047111  EM55:   .ASCII  /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
12836  061254  020107  051117  051040
12837  061262  040505  044504  043516
12838  061270  053440  052111  020110
12839  061276  054105  042520  052103
12840  061304  042105  040440  042104
12841  061312  042522  051523  047440
12842  061320  042526  043122  047514
12843  061326  020127  051105  047522
12844  061334  006522     012
12845  061337     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
12846  061344  020104  046511  051120
```

```
12847  061352  050117  051105  051040
12848  061360  043505  051511  042524
12849  061366  020122  044103  047101
12850  061374  042507  005015
12851  061400  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12852  061406  052101  020101  044507
12953  061414  042526  020123  044127
12854  061422  052101  051440  047510
12855  061430  046125  020104  042502
12856  061436  052040  042510  042522
12857  061444  005015
12858  061446  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
12859  061454  042105  042040  052101
12860  061462  020101  044507  042526
12861  061470  020123  042522  044507
12862  061476  052123  051105  041440
12863  061504  047117  042524  052116
12864  061512  020123  043101  042524
12865  061520  020122  050117  051105
12866  061526  052101  047511  000116
12867  061534  040504  040524  051040  EM56:   .ASCII  /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
12868  061542  040505  020104  044527
12869  061550  044124  040440  020116
12870  061556  054105  042520  052103
12871  061564  042105  040440  042104
12872  061572  042522  051523  047440
12873  061600  042526  043122  047514
12874  061606  020127  051105  047522
12875  061614  020122  051511  044440
12876  061622  041516  051117  042522
12877  061630  052103  005015
12878  061634  047527  042122  047040          .ASCII  /WORD NO. 1 TO 260 SHOULD BE READ.- WORD NO 261 TO 266 SHOULD/<15><12>
12879  061642  027117  030440  052040
12880  061650  020117  033062  020060
12881  061656  044123  052517  042114
12882  061664  041040  020105  042522
12883  061672  042101  020054  047527
12884  061700  042122  047040  020117
12885  061706  033062  020061  047524
12886  061714  031040  033066  051440
12887  061722  047510  046125  006504
12888  061730     012
12889  061731     102  020105  044103          .ASCIZ  /BE CHANGED/
12890  061736  047101  042507  000104
12891  061744  052101  042524  050115  EM57:   .ASCII  /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
12892  061752  044524  043516  042040
12893  061760  052101  020101  047503
12894  061766  046515  047101  020104
12895  061774  044527  044124  053440
12896  062002  047522  043516  043040
12897  062010  051117  040515  020124
12898  062016  044502  020124  040503
12899  062024  051525  042105  005015
12900  062032  046511  051120  050117          .ASCII  /IMPROPER REGISTER CHANGE/<15><12>
```

```
12901  062040  051105  051040  043505
12902  062046  051511  042524  020122
12903  062054  044103  047101  042507
12904  062062  005015
12905  062064  047507  042117  042040          .ASCII   GOOD DATA GIVES WHAT SHOULD BE THERE<15><12>
12906  062072  052101  020101  044507
12907  062100  042526  020123  044127
12908  062106  052101  051440  047510
12909  062114  046125  020104  042502
12910  062122  052040  042510  042522
12911  062130  005015
12912  062132  042522  042503  053111          .ASCIZ   RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER<
12913  062140  043105  042040  052101
12914  062146  020101  044507  042526
12915  062154  020123  042522  044507
12916  062162  052123  051105  041440
12917  062170  047117  042524  052116
12918  062176  020123  043101  042524
12919  062204  020122  052101  042524
12920  062212  050115  042524  020104
12921  062220  040504  040524  052040
12922  062226  040522  051516  042506
12923  062234  000122
12924
12925  062236  052101  042524  050115  EMSG:   .ASCII   ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER<15><
12926  062244  044524  043516  052040
12927  062252  020117  047515  044504
12928  062260  054506  051040  043505
12929  062266  051511  042524  020122
12930  062274  052504  044522  043516
12931  062302  040440  020116  050117
12932  062310  051105  052101  047511
12933  062316  020116  040503  051525
12934  062324  042105  044440  050115
12935  062332  047522  042520  006522
12936  062340     012
12937  062341     122  043505  051511          .ASCII   <REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE<15><12>
12938  062346  042524  020122  044103
12939  062354  047101  042507  020056
12940  062362  047507  042117  042040
12941  062370  052101  020101  044507
12942  062376  042526  020123  044127
12943  062404  052101  051440  047510
12944  062412  046125  020104  042502
12945  062420  052040  042510  042522
12946  062426  005015
12947  062430  042522  042503  053111          .ASCII   RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED <15
12948  062436  042105  042040  052101
12949  062444  020101  044507  042526
12950  062452  020123  042522  044507
12951  062460  052123  051105  041440
12952  062466  047117  042524  052116
12953  062474  020123  043101  042524
12954  062502  020122  050117  051105
```

```
12955  062510  052101  047511  020116
12956  062516  040527  020123  052101
12957  062524  042524  050115  042524
12958  062532  006504     012
12959  062535     115  042117  043111          .ASCIZ   MODIFYING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERR
12960  062542  044531  043516  051040
12961  062550  043505  043440  053111
12962  062556  051505  040440  042104
12963  062564  042522  051523  047440
12964  062572  020106  042522  044507
12965  062600  052123  051105  041040
12966  062606  044505  043516  046440
12967  062614  042117  043111  042511
12968  062622  020104  044127  041511
12969  062630  020110  040503  051525
12970  062636  042105  042440  051122
12971  062644  051117     000
12972  062647     122  041510  030523  EM61:   .ASCIZ  /RHOS1 HAS SOME INCORRECT STATUS BITS = 1, OR = 0/
12973  062654  044040  051501  051440
12974  062662  046517  020105  047111
12975  062670  047503  051122  041505
12976  062676  020124  052123  052101
12977  062704  051525  041040  052111
12978  062712  020123  020075  026061
12979  062720  047440  020122  020075
12980  062726  000060
12981  062730  044122  051504  020061  EM62:   .ASCIZ  /RHOS1 HAS SOME INCORRECT STATUS BITS = 1, OR = 0/
12982  062736  040510  020123  047523
12983  062744  042515  044440  041516
12984  062752  051117  042522  052103
12985  062760  051440  040524  052524
12986  062766  020123  044502  051524
12987  062774  036440  030440  020054
12988  063002  051117  036440  030040
12989  063010     000
12990  063011     122  042110  030523  EM63:   .ASCIZ  /RHOS1 CONTENTS DURING COMMAND WERE IN ERROR/
12991  063016  041440  047117  042524
12992  063024  052116  020123  052504
12993  063032  044522  043516  041440
12994  063040  046517  040515  042116
12995  063046  053440  051105  020105
12996  063054  047111  042440  051122
12997  063062  051117     000
12998  063065     122  041505  046101  EM64:   .ASCII  /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE <15><12>/
12999  063072  041111  040522  042524
13000  063100  041440  046517  040515
13001  063106  042116  041440  052501
13002  063114  042523  020104  046511
13003  063122  051120  050117  051105
13004  063130  051040  043505  051511
13005  063136  042524  020122  044103
13006  063144  047101  042507  005015
13007  063152  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE <15><12>/
13008  063160  052101  020101  044507
```

```
13009  063166  042526  020123  044127
13010  063174  052101  051440  047510
13011  063202  046125  020104  042502
13012  063210  052040  042510  042522
13013  063216  005015
13014  063220  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/<15><12>
13015  063226  042105  042040  052101
13016  063234  020101  044507  042526
13017  063242  020123  042522  044507
13018  063250  052123  051105  041440
13019  063256  047117  042524  052116
13020  063264  020123  043101  042524
13021  063272  020122  047503  046515
13022  063300  047101  000104
13023  063304  047111  042524  051122  EM65:   .ASCIZ  /INTERRUPT FAILING/<15><12>
13024  063312  050125  020124  040506
13025  063320  046111  047111  000107
13026  063326  042510  042101  051105  EM66:   .ASCII  /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST CAUSED/<15><12>
13027  063334  040440  042116  042040
13028  063342  052101  020101  047503
13029  063350  046515  047101  020104
13030  063356  047506  020122  042510
13031  063364  042101  051440  046105
13032  063372  041505  044524  047117
13033  063400  052040  051505  020124
13034  063406  040503  051525  042105
13035  063414  005015
13036  063416  051105  047522  020122          .ASCII  /ERROR - RHOST GIVES TRACK BEING WRITTEN OR READ ON CYL 0, SCTR 0/<15><12>
13037  063424  020055  044122  051504
13038  063432  020124  044507  042526
13039  063440  020123  051124  041501
13040  063446  020113  042502  047111
13041  063454  020107  051127  052111
13042  063462  042524  020116  051117
13043  063470  051040  040505  020104
13044  063476  047117  041440  046131
13045  063504  030040  020054  041523
13046  063512  051124  030040  005015
13047  063520  042522  042101  044040  EM67:   .ASCII  /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST. /<15><12>
13048  063526  040505  042504  020122
13049  063534  047101  020104  040504
13050  063542  040524  042440  051122
13051  063550  051117  044440  020116
13052  063556  042510  042101  051440
13053  063564  046105  041505  044524
13054  063572  047117  052040  051505
13055  063600  027124  006412
13056  063604  044506  051522  020124          .ASCII  /FIRST FOUR WORD NUMBERS ARE THE HEADER. /<15><12>
13057  063612  047506  051125  053440
13058  063620  051117  020104  052516
13059  063626  041115  051105  020123
13060  063634  051101  020105  044124
13061  063642  020105  042510  042101
13062  063650  051105  005056      015
```

```
 13063  063655     127  051117  020104          .ASCII  /WORD NUMBERS 5 TO 260 ARE DATA WORDS./<12><15>
 13064  063662  052516  041115  051105
 13065  063670  020123  020065  047524
 13066  063676  031040  030066  040440
 13067  063704  042522  042040  052101
 13068  063712  020101  047527  042122
 13069  063720  026123  006412
 13070  063724  047101  020104  047111          .ASCIZ   AND IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER./
 13071  063732  042040  052101  020101
 13072  063740  047527  042122  020123
 13073  063746  044502  051524  032040
 13074  063754  032454  033054  033454
 13075  063762  034054  043440  053111
 13076  063770  020105  051124  041501
 13077  063776  020113  052516  041115
 13078  064004  051105  000056
 13079
 13080  064010  042522  042101  044040  EMTO:   .ASCII  /READ HEADER AND DATA ERROR IN/<15><12>
 13081  064016  040505  042504  020122
 13082  064024  047101  020104  040504
 13083  064032  040524  042440  051122
 13084  064040  051117  044440  006516
 13085  064046     012
 13086  064047     104  043111  042506          .ASCII  /DIFFERENCE LINE TEST/<15><12>
 13087  064054  042522  041516  020105
 13088  064062  044514  042516  052040
 13089  064070  051505  006524     012
 13090  064075     127  051117  020104          .ASCII  /WORD NOS 1-4 GIVE HEADER/<15><12>
 13091  064102  047516  020123  026461
 13092  064110  020064  044507  042526
 13093  064116  044040  040505  042504
 13094  064124  006522     012
 13095  064127     127  051117  020104          .ASCIZ  /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS
 13096  064134  047516  020123  026465
 13097  064142  033062  020060  044507
 13098  064150  042526  042040  052101
 13099  064156  020101  044127  041511
 13100  064164  020110  051511  052040
 13101  064172  042510  041440  046131
 13102  064200  047111  042504  020122
 13103  064206  042101  051104  051505
 13104  064214  000123
 13105  064216  047506  041522  047111  EM7I:   .ASCII  /FORCING OPI BY 3 INDEX PULSES <15><12
 13106  064224  020107  050117  020111
 13107  064232  054502  031440  044440
 13108  064240  042116  054105  050040
 13109  064246  046125  042523  006523
 13110  064254     012
 13111  064255     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE  15  12
 13112  064262  020104  046511  051120
 13113  064270  050117  051105  051040
 13114  064276  043505  051511  042524
 13115  064304  020122  044103  047101
 13116  064312  042507  000015
```

```
13117  064316  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE /<15><12>
13118  064324  052101  020101  044507
13119  064332  042526  020123  044127
13120  064340  052101  051440  047510
13121  064346  046125  020104  042502
13122  064354  052040  042510  042522
13123  064362  005015
13124  064364  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
13125  064372  042105  042040  052101
13126  064400  020101  044507  042526
13127  064406  020123  042522  044507
13128  064414  052123  051105  041440
13129  064422  047117  042524  052116
13130  064430  020123  043101  042524
13131  064436  020122  020063  047111
13132  064444  042504  020130  052520
13133  064452  051514  051505     000
13134  064457     127  044510  042514  EM72:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13135  064464  052440  044523  043516
13136  064472  052440  044516  052502
13137  064500  020123  006502     012
13138  064505     122  040505  020104          .ASCII   READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13139  064512  040504  040524  041440
13140  064520  052501  042523  020104
13141  064526  046511  051120  050117
13142  064534  051105  051040  043505
13143  064542  051511  042524  020122
13144  064550  044103  047101  042507
13145  064556  005015
13146  064560  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE /<15><12>
13147  064566  052101  020101  044507
13148  064574  042526  020123  044127
13149  064602  052101  051440  047510
13150  064610  046125  020104  042502
13151  064616  052040  042510  042522
13152  064624  005015
13153  064626  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13154  064634  042105  042040  052101
13155  064642  020101  044507  042526
13156  064650  020123  044127  052101
13157  064656  053440  051501  052040
13158  064664  042510  042522  040440
13159  064672  052106  051105  041440
13160  064700  046517  040515  042116
13161  064706     000
13162  064707     127  044510  042514  EM73:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13163  064714  052440  044523  043516
13164  064722  052440  044516  052502
13165  064730  020123  006502     012
13166  064735     122  040505  020104          .ASCIZ  /READ DATA INCORRECT/
13167  064742  040504  040524  044440
13168  064750  041516  051117  042522
13169  064756  052103     000
13170  064761     127  044510  042514  EM74:   .ASCII   WHILE USING UNIBUS B /<15><12>
```

```
13171  064766  052440  044523  043516
13172  064774  052440  044516  052502
13173  065002  020123  006502     012
13174  065007     127  044522  042524          .ASCII  /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13175  065014  042040  052101  020101
13176  065022  047503  046515  047101
13177  065030  020104  040503  051525
13178  065036  042105  044440  050115
13179  065044  047522  042520  020122
13180  065052  042522  044507  052123
13181  065060  051105  041440  040510
13182  065066  043516  006505     012
13183  065073     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13184  065100  040504  040524  043440
13185  065106  053111  051505  053440
13186  065114  040510  020124  044123
13187  065122  052517  042114  041040
13188  065130  020105  044124  051105
13189  065136  006505     012
13190  065141     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13191  065146  042526  020104  040504
13192  065154  040524  043440  053111
13193  065162  051505  051040  043505
13194  065170  051511  042524  020122
13195  065176  047503  052116  047105
13196  065204  051524  040440  052106
13197  065212  051105  041440  046517
13198  065220  040515  042116     000
13199  065225     127  044510  042514  EM75:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13200  065232  052440  044523  043516
13201  065240  052440  044516  052502
13202  065246  020123  006502     012
13203  065253     127  044522  042524          .ASCIZ  /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
13204  065260  042040  052101  020101
13205  065266  047503  046515  047101
13206  065274  020104  044103  047101
13207  065302  042507  020104  051127
13208  065310  052111  020105  051106
13209  065316  046517  041040  043125
13210  065324  042506  000122
13211  065330  044127  046111  020105  EM76:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13212  065336  051525  047111  020107
13213  065344  047125  041111  051525
13214  065352  041040  005015
13215  065356  051127  052111  020105          .ASCII  /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
13216  065364  044103  041505  020113
13217  065372  040503  051525  042105
13218  065400  044440  050115  047522
13219  065406  042520  020122  042522
13220  065414  044507  052123  051105
13221  065422  041440  040510  043516
13222  065430  006505     012
13223  065433     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13224  065440  040504  040524  043440
```

```
13225  065446  053111  051505  053440
13226  065454  040510  020124  044123
13227  065462  052517  042114  041040
13228  065470  020105  044124  051105
13229  065476  006505     012
13230  065501     122  041505  044505          .ASCIZ  RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13231  065506  042526  020104  040504
13232  065514  040524  043440  053111
13233  065522  051505  051040  043505
13234  065530  051511  042524  020122
13235  065536  047503  052116  047105
13236  065544  051524  040440  052106
13237  065552  051105  041440  046517
13238  065560  040515  042116     000
13239  065565     120  042522  047514  EM77:   .ASCII  /PRELOADING 'RHCC' PRIOR TO DOING NEXT TEST DOES NOT PRODUCE CORRECT RES
13240  065572  042101  047111  020107
13241  065600  051047  041510  023503
13242  065606  050040  044522  051117
13243  065614  052040  020117  047504
13244  065622  047111  020107  042516
13245  065630  052130  052040  051505
13246  065636  020124  047504  051505
13247  065644  047040  052117  050040
13248  065652  047522  052504  042503
13249  065660  041440  051117  042522
13250  065666  052103  051040  051505
13251  065674  046125  006524     012
13252  065701     124  042510  042522          .ASCIZ  /THEREFORE NEXT TEST RESULTS ARE SUSPECT WITH REGARD TO 'RHCC' CONTENTS/
13253  065706  047506  042522  047040
13254  065714  054105  020124  042524
13255  065722  052123  051040  051505
13256  065730  046125  051524  040440
13257  065736  042522  051440  051525
13258  065744  042520  052103  053440
13259  065752  052111  020110  042522
13260  065760  040507  042122  052040
13261  065766  020117  051047  041510
13262  065774  023503  041440  047117
13263  066002  042524  052116  000123
13264
13265  066010  042101  051104  051505  EM100:  .ASCIZ  /ADDRESS PLUG CHANGE RESULTED IN BAD REGISTER DATA/
13266  066016  020123  046120  043525
13267  066024  041440  040510  043516
13268  066032  020105  042522  052523
13269  066040  052114  042105  044440
13270  066046  020116  040502  020104
13271  066054  042522  044507  052123
13272  066062  051105  042040  052101
13273  066070  000101
13274  066072  047125  052111  042040  EM101:  .ASCIZ  /UNIT DID NOT GO OFFLINE WHEN ADDRESS PLUG REMOVED
13275  066100  042111  047040  052117
13276  066106  043440  020117  043117
13277  066114  046106  047111  020105
13278  066122  044127  047105  040440
```

```
13279  066130  042104  042522  051523
13280  066136  050040  052514  020107
13281  066144  042522  047515  042526
13282  066152  000104
13283  066154  047125  052111  047040  EM102:  .ASCIZ  /UNIT NOT AVAILABLE AFTER ADDRESS PLUG REPLACED/
13284  066162  052117  040440  040526
13285  066170  046111  041101  042514
13286  066176  040440  052106  051105
13287  066204  040440  042104  042522
13288  066212  051523  050040  052514
13289  066220  020107  042522  046120
13290  066226  041501  042105     000
13291  066233     122  043505  051511  EM103:  .ASCIZ  /REGISTER CONTENTS INCORRECT BEFORE A DIAG MODE SEEK/
13292  066240  042524  020122  047503
13293  066246  052116  047105  051524
13294  066254  044440  041516  051117
13295  066262  042522  052103  041040
13296  066270  043105  051117  020105
13297  066276  020101  044504  043501
13298  066304  046440  042117  020105
13299  066312  042523  045505     000
13300  066317     122  043505  051511  EM104:  .ASCIZ  /REGISTER CONTENTS INCORRECT AFTER A DIAG MODE SEEK/
13301  066324  042524  020122  047503
13302  066332  052116  047105  051524
13303  066340  044440  041516  051117
13304  066346  042522  052103  040440
13305  066354  052106  051105  040440
13306  066362  042040  040511  020107
13307  066370  047515  042504  051440
13308  066376  042505  000113
```

```
13309
13310
13311  066402  040506  040524  020114  CPHALT: .ASCII   FATAL ERROR - SEE DOCUMENT LISTING/<15><12><15><12>
13312  066410  051105  047522  020122
13313  066416  020055  042523  020105
13314  066424  047504  052503  042515
13315  066432  052116  046040  051511
13316  066440  044524  043516  005015
13317  066446  005015
13318  066450  044124  020105  047503          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>
13319  066456  052116  047522  046114
13320  066464  051105  047440  020122
13321  066472  042504  044526  042503
13322  066500  044040  051501  043440
13323  066506  047117  020105  043117
13324  066514  046106  047111  026105
13325  066522  046040  051517  005524
13326  066530     012
13327  066531     047  042522  042101          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
13328  066536  023531  020054  042502
13329  066544  047503  042515  052440
13330  066552  040516  040526  046111
13331  066560  041101  042514  020054
13332  066566  051117  044040  051501
13333  066574  051440  040524  052524
13334  066602  020123  044502  051524
13335  066610  005015
13336  066612  044127  041511  020110          .ASCIZ  /WHICH CANNOT BE CLEARED/
13337  066620  040503  047116  052117
13338  066626  041040  020105  046103
13339  066634  040505  042522  000104
13340
13341
13342
13343
13344  066642  041520  020040  020040  DH1:    .ASCII  /PC      TEST    WAT     BIT     REG     REG     RHCS1/<15><12>
13345  066650  020040  042524  052123
13346  066656  020040  020040  040527
13347  066664  020124  020040  020040
13348  066672  044502  020124  020040
13349  066700  020040  042522  020107
13350  066706  020040  020040  042522
13351  066714  020107  020040  020040
13352  066722  044122  051503  006461
13353  066730     012
13354  066731     040  020040  020040          .ASCIZ  /       NO      PC      EXPECT  ADDRESS CONTENT CONTENT/
13355  066736  020040  047040  020117
13356  066744  020040  020040  050040
13357  066752  020103  020040  020040
13358  066760  042440  050130  041505
13359  066766  020124  040440  042104
13360  066774  042522  051523  041440
13361  067002  047117  042524  052116
13362  067010  041440  047117  042524
```

```
13363  067016  052116     000
13364  067021     120  020103  020040  DH4:    .ASCII  /PC      TEST    WAT     BIT     REG     TIME IN/<15><12>
13365  067026  020040  052040  051505
13366  067034  020124  020040  053440
13367  067042  052101  020040  020040
13368  067050  041040  052111  020040
13369  067056  020040  051040  043505
13370  067064  020040  020040  052040
13371  067072  046511  020105  047111
13372  067100  005015
13373  067102  020040  020040  020040          .ASCIZ  /       NO      PC      EXPECT  ADDRESS 10 MSEC/
13374  067110  020040  047516  020040
13375  067116  020040  020040  041520
13376  067124  020040  020040  020040
13377  067132  054105  042520  052103
13378  067140  020040  042101  051104
13379  067146  051505  020123  030061
13380  067154  046440  042523  000103
13381  067162  041520  020040  020040  DH5:    .ASCII  /PC      TEST    REG     GOOD    RECEIVED/<15><12>
13382  067170  020040  042524  052123
13383  067176  020040  020040  042522
13384  067204  020107  020040  020040
13385  067212  047507  042117  020040
13386  067220  020040  042522  042503
13387  067226  053111  042105  005015
13388  067234  020040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA/
13389  067242  020040  047516  020040
13390  067250  020040  020040  042101
13391  067256  051104  051505  020123
13392  067264  040504  040524  020040
13393  067272  020040  040504  040524
13394  067300     000
13395  067301     120  020103  020040  DH6:    .ASCII  /PC      TEST    REG     RECEIVED/<15><12>
13396  067306  020040  052040  051505
13397  067314  020124  020040  051040
13398  067322  043505  020040  020040
13399  067330  051040  041505  044505
13400  067336  042526  006504     012
13401  067343     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA/
13402  067350  020040  047040  020117
13403  067356  020040  020040  040440
13404  067364  042104  042522  051523
13405  067372  042040  052101  000101
13406  067400  041520  020040  020040  DH7:    .ASCIZ  /PC      TEST    REG     ADDRESS/
13407  067406  020040  042524  052123
13408  067414  020040  020040  042522
13409  067422  020107  020040  020040
13410  067430  042101  051104  051505
13411  067436  000123
13412
13413  067440  041520  020040  020040  DH10:   .ASCII  /PC      TEST    FAILING CONTENT CONTENT CONTENT CONTENT/<15><12>
13414  067446  020040  042524  052123
13415  067454  020040  020040  040506
13416  067462  046111  047111  020107
```

```
13417  067470  047503  052116  047105
13418  067476  020124  047503  052116
13419  067504  047105  020124  047503
13420  067512  052116  047105  020124
13421  067520  047503  052116  047105
13422  067526  006524     012
13423  067531     040  020040  020040          .ASCIZ  /       NO      REG.    RHCS1   RHCS1   RHDS1   RHER1/
13424  067536  020040  047040  020117
13425  067544  020040  020040  051040
13426  067552  043505  020056  020040
13427  067560  051040  041510  030523
13428  067566  020040  051040  041510
13429  067574  030523  020040  051040
13430  067602  042110  030523  020040
13431  067610  051040  042510  030522
13432  067616     000
13433  067617     120  020103  020040  DH26:   .ASCII  /PC     TEST    CONT OF CONT OF CONT OF CONT OF CONT OF CONT OF/<15><12
13434  067624  020040  052040  051505
13435  067632  020124  020040  041440
13436  067640  047117  020124  043117
13437  067646  041440  047117  020124
13438  067654  043117  041440  047117
13439  067662  020124  043117  041440
13440  067670  047117  020124  043117
13441  067676  041440  047117  020124
13442  067704  043117  041440  047117
13443  067712  020124  043117  005015
13444  067720  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHCS2   RHDS1   RHER1   RHER2   RHER3
13445  067726  020040  047516  020040
13446  067734  020040  020040  044122
13447  067742  051503  020061  020040
13448  067750  044122  051503  020062
13449  067756  020040  044122  051504
13450  067764  020061  020040  044122
13451  067772  051105  020061  020040
13452  070000  044122  051105  020062
13453  070006  020040  044122  051105
13454  070014  000063
13455  070016  041520  020040  020040  DH30:   .ASCII  /PC     TEST    WORD    GOOD    BAD/<15><12>
13456  070024  020040  042524  052123
13457  070032  020040  020040  047527
13458  070040  042122  020040  020040
13459  070046  047507  042117  020040
13460  070054  020040  040502  006504
13461  070062     012
13462  070063     040  020040  020040          .ASCIZ  /       NC      NO      DATA    DATA/<15><12>
13463  070070  020040  047040  020117
13464  070076  020040  020040  047040
13465  070104  020117  020040  020040
13466  070112  042040  052101  020101
13467  070120  020040  042040  052101
13468  070126  006501  000012
13469
13470  070132  041520  020040  020040  DH51:   .ASCII  /PC     TEST    REG     GOOD    RECVD   ILLEGL<15><12>
```

```
13471  070140  020040  042524  052123
13472  070146  020040  020040  042522
13473  070154  020107  020040  020040
13474  070162  047507  042117  020040
13475  070170  020040  042522  053103
13476  070176  020104  020040  046111
13477  070204  042514  046107  005015
13478  070212  020040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    FUNCTN/
13479  070220  020040  047516  020040
13480  070226  020040  020040  042101
13481  070234  051104  051505  020123
13482  070242  040504  040524  020040
13483  070250  020040  040504  040524
13484  070256  020040  020040  052506
13485  070264  041516  047124     000
13486
13487  070271     120  020103  020040  DH60:   .ASCII  /PC     TEST    REG     GOOD    RECVD   MODFING/<15><12>
13488  070276  020040  052040  051505
13489  070304  020124  020040  051040
13490  070312  043505  020040  020040
13491  070320  043440  047517  020104
13492  070326  020040  051040  041505
13493  070334  042126  020040  046440
13494  070342  042117  044506  043516
13495  070350  005015
13496  070352  020040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    REG/
13497  070360  020040  047516  020040
13498  070366  020040  020040  042101
13499  070374  051104  051505  020123
13500  070402  040504  040524  020040
13501  070410  020040  040504  040524
13502  070416  020040  020040  042522
13503  070424  000107
13504  070426  041520  020040  020040  DH61:   .ASCII  /PC     TEST    PC OF   RHCS1/<15><12>
13505  070434  020040  042524  052123
13506  070442  020040  020040  041520
13507  070450  047440  020106  020040
13508  070456  044122  051503  006461
13509  070464     012
13510  070465     040  020040  020040          .ASCIZ  /       NO      JSR     WAS/
13511  070472  020040  047040  020117
13512  070500  020040  020040  045040
13513  070506  051123  020040  020040
13514  070514  053440  051501     000
13515  070521     120  020103  020040  DH62:   .ASCII  /PC     TEST    PC OF   RHDS1/<15><12>
13516  070526  020040  052040  051505
13517  070534  020124  020040  050040
13518  070542  020103  043117  020040
13519  070550  051040  042110  030523
13520  070556  005015
13521  070560  020040  020040  020040          .ASCIZ  /       NO      JSR     WAS/
13522  070566  020040  047516  020040
13523  070574  020040  020040  051512
13524  070602  020122  020040  020040
```

```
13525  070610  040527  000123
13526  070614  041520  020040  020040  DH65:   .ASCII  /PC      TEST    CONT    CONT    CONT    /<15><12>
13527  070622  020040  042524  052123
13528  070630  020040  020040  047503
13529  070636  052116  020040  020040
13530  070644  047503  052116  020040
13531  070652  020040  047503  052116
13532  070660  020040  020040  005015
13533  070666  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1   /
13534  070674  020040  047516  020040
13535  070702  020040  020040  044122
13536  070710  051503  020061  020040
13537  070716  044122  051501  020040
13538  070724  020040  044122  051504
13539  070732  020061  020040     000
13540  070737     120  020103  020040  DH66:   .ASCIZ  /PC      TEST    RHDST   RHER1   RHER2   RHER3   RHCS1   RHCS2/
13541  070744  020040  052040  051505
13542  070752  020124  020040  051040
13543  070760  042110  052123  020040
13544  070766  051040  042510  030522
13545  070774  020040  051040  042510
13546  071002  031122  020040  051040
13547  071010  042510  031522  020040
13548  071016  051040  041510  030523
13549  071024  020040  051040  041510
13550  071032  031123     000
13551
13552  071035     120  020103  020040  DH77:   .ASCII  /PC      TEST    PC OF   REG.    GOOD    BAD/<15><12>
13553  071042  020040  052040  051505
13554  071050  020124  020040  050040
13555  071056  020103  043117  020040
13556  071064  051040  043505  020056
13557  071072  020040  043440  047517
13558  071100  020104  020040  041040
13559  071106  042101  005015
13560  071112  020040  020040  020040          .ASCIZ  /       NO      JSR     ADDR    DATA    DATA/
13561  071120  020040  047516  020040
13562  071126  020040  020040  051512
13563  071134  020122  020040  020040
13564  071142  042101  051104  020040
13565  071150  020040  040504  040524
13566  071156  020040  020040  040504
13567  071164  040524     000
13568
13569          071170                  .EVEN
13570
13571  071170  001116  004604  041616  DT1:    .WORD   $ERRPC,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,CS1,0
13572  071176  041622  041620  001126
13573  071204  002362  000000
13574  071210  001116  004604  041616  DT4:    .WORD   $ERRPC,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,WAITTM,0
13575  071216  041622  041620  001126
13576  071224  041624  000000
13577  071230  001116  004604  004600  DT5:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
13578  071236  001124  001126  000000
```

```
PRINCE-11-DCRJI-A, RPC4 5 6 FUNCT. CONT. TST-PT 1
DCRJIA.P11      POWER DOWN AND UP ROUTINES

 13578  071244  001116  004604  004600  DT6:    .WORD   SERRPC,TSTNM,REGADR,SBDDAT,0
 13579  071252  001126  000000
 13580  071256  001116  004604  001200  DT7:    .WORD   SERRPC,TSTNM,STMP1,0
 13581  071264  000000
 13582
 13583  071266  001116  004604  001122  DT10:   .WORD   SERRPC,TSTNM,SBDADR,CS1,CS2,DS1,ER1,0
 13584  071274  002362  002360  002404
 13585  071302  002364  000000
 13586  071306  001116  004604  002362  DT26:   .WORD   SERRPC,TSTNM,CS1,CS2,DS1,ER1,ER2,ER3,0
 13587  071314  002360  002404  002364
 13588  071322  002370  002376  000000
 13589  071330  001116  004604  004602  DT30:   .WORD   SERRPC,TSTNM,ERWORD,SGDDAT,SBDDAT,0
 13590  071336  001124  001126  000000
 13591
 13592  071344  001116  004604  004600  DT51:   .WORD   SERRPC,TSTNM,REGADR,SGDDAT,SBDDAT,ILLEGL,0
 13593  071352  001124  001126  002464
 13594  071360  000000
 13595
 13596  071362  001116  004604  004600  DT60:   .WORD   SERRPC,TSTNM,REGADR,SGDDAT,SBDDAT,SBDADR,0
 13597  071370  001124  001126  001122
 13598  071376  000000
 13599  071400  001116  004604  041422  DT61:   .WORD   SERRPC,TSTNM,PCJSR,SBDADR,0
 13600  071406  001122  000000
 13601  071412  001116  004604  041422  DT62:   .WORD   SERRPC,TSTNM,PCJSR,SBDADR,0
 13602  071420  001122  000000
 13603  071424  001116  004604  002362  DT65:   .WORD   SERRPC,TSTNM,CS1,AS,DS1,0
 13604  071432  002400  002404  000000
 13605  071440  001116  004604  002364  DT66:   .WORD   SERRPC,TSTNM,ER1,ER2,ER3,CS1,CS2,0
 13606  071446  002370  002376  002362
 13607  071454  002360  000000
 13608
 13609  071460  001116  004604  041422  DT77:   .WORD   SERRPC,TSTNM,PCJSR,REGADR,SGDDAT,SBDDAT,0
 13610  071466  004600  001124  001126
 13611  071474  000000
 13612
 13613  071476     000     000     000  DF1:    .BYTE   0,0,0,0,0,0,0
 13614  071501     000     000     000
 13615  071504     000
 13616  071505     000     000     000  DF4:    .BYTE   0,0,0,0,0,1,0
 13617  071510     000     000     001
 13618  071513     000
 13619  071514     000     000     000  DF5:    .BYTE   0,0,0,0,0
 13620  071517     000     000
 13621  071521     000     000     000  DF6:    .BYTE   0,0,0,0
 13622  071524     000
 13623  071525     000     000     000  DF7:    .BYTE   0,0,0
 13624
 13625  071530     000     000     000  DF10:   .BYTE   0,0,0,0,0,0,0
 13626  071533     000     000     000
 13627  071536     000
 13628
 13629  071537     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0,0
 13630  071542     000     000     000
 13631  071545     000     000
```

```
:3633
13634  071547  000  000  000  DF30:  .BYTE  0,0,0,0,0
13635  071552  000  000
13636
13637  071554  000  000  000  DF51:  .BYTE  0,0,0,0,0,0
13638  071557  000  000  000
13639
13640  071562  000  000  000  DF60:  .BYTE  0,0,0,0,0,0
13641  071565  000  000  000
13642  071566  000  000  000  DF61:  .BYTE  0,0,0,0
13643  071573  000  000
13644  071574  000  000  000  DF62:  .BYTE  0,0,0,0
13645  071577  000  000
13646  071600  000  000  000  DF65:  .BYTE  0,0,0,0,0
13647  071603  000  000
13648  071604  000  000  000  DF66:  .BYTE  0,0,0,0,0,0,0,0
13649  071611  000  000  000
13650  071613  000  000
13651
13652  071615  000  000  000  DF77:  .BYTE  0,0,0,0,0,0
13653  071620  000  000  000
13654
13655          071624              .EVEN
13656
13657
13658          000001              .END
```

DCRJXA.P11 CROSS REFERENCE TABLE

```
...E    = 020000          1299#   2728    2795
...     = 000040          1297#
...     = 100000          1257#
...E    = 001000          1179#
...     = 002400          1395#  10831*  11754   13604
...A    = 100000          1167#   2710    2779    3220    4260    4559    4829    4949    8385    8637   10040   10526
...ABLE   004666          1464#   1976
...ENT    004740          1488#   1913*   1976*   2730    2804    3212    3406    3590    3762    4063    4251    4582    4331
                          4951    4951    8376    9628   10044
...0    = 000001          1199#
...1    = 000002          1200#
...2    = 000004          1201#
...3    = 000010          1202#
...4    = 000020          1203#
...5    = 000040          1204#
...6    = 000100          1205#
...7    = 000200          1206#
...8    = 000400          1142#
...9    = 001000          1143#
...     = 002356          1394#  11682
...I    = 000010          1113#   3360    3381
...SECH   043522          1711   11041#
...GIN    005012           192    1519#  12029
...GIN1   004762           196    1513#
...GIN2   004776           194    1516#
...RD     020072          4118    4140#
...0    = 000001           152#
...00   = 000001           142#    152
...01   = 000002           141#    151
...02   = 000004           140#    150
...03   = 000010           139#    149
...04   = 000020           138#    148
...05   = 000040           137#    147
...06   = 000100           136#    146
...07   = 000200           135#    145
...08   = 000400           134#    144   11144
...09   = 001000           133#    143   11152   11634
...1    = 000002           151#
...10   = 002000           132#  11618
...11   = 004000           131#  11159
...12   = 010000           130#
...13   = 020000           129#  11625
...14   = 040000           128#  11130
...15   = 100000           127#   4956
...2    = 000004           150#
...3    = 000010           149#
...4    = 000020           148#  10613
...5    = 000040           147#
...6    = 000100           146#
...7    = 000200           145#
...8    = 000400           144#
...9    = 001000           143#
...VEC  = 000014           159#
...     = 002374          1393#  11746
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
CAT      020036          4098   4121#
CC       002416          1402#  11786
CHECK    041424          2296   2401   3537   3730   4359   4618   4673   10506#
CHECKC   041462          10510  10516#
CHECKT   041446          2591   2905   3044   3160   3260   3937   4158   4331   4384   4687   4859   5006   5153
                         5308   5470   5664   5829   6016   6170   6365   6530   6743   6933   7175   7342   7509
                         7760   8007   8171   8464   8722   8805   8964   9059   9178   9296   9425   9552   9689
                         9860   10512#
CHREG    042246          2338   2468   2692   2705   2716   2723   2763   2774   2783   2790   2797   3215   3253
                         3364   3374   3429   3652   3663   3824   3835   4255   4262   4564   4574   4817   4824
                         4937   4944   8380   8387   8632   8639   9914   10016  10028  10035  10181  10751#
CKSUM =  104406          11129  11610  11633  11986#
CLAREA   041236          5105   5124   5273   5436   5610   5615   5625   5630   5777   5782   5790   5795   5972
                         5982   6128   6136   6311   6316   6326   6331   6478   6483   6491   6496   6687   6632
                         6702   6707   6879   6884   6892   6897   7092   7109   7128   7141   7289   7307   7505
                         7522   7541   7554   7709   7726   7953   7972   8112   8130   8135   8419   8434   8439
                         8676   8692   8697   8859   8864   8869   8874   8879   8909   8922   8930   9339   9344
                         9349   9354   9359   9388   9393   9600   9605   9610   9615   9620   9648   9653   9836
                         9952   10081  10088  10404#
CLDISK   041366          1909   2061   2096   2135   2195   2243   2292   2397   2523   2542   2587   2755   2901
                         2952   3040   3111   3256   3492   3533   3726   3899   3908   3933   4111   4130   4154
                         4327   4355   4669   4683   4976   5002   5034   5087   5269   5429   5603   5773   5965
                         6124   6304   6474   6671   6875   7076   7285   7489   7705   7888   7935   8109   8304
                         8528   8672   8801   8841   9028   9146   9292   9332   9541   9593   9829   9948   10075
                         10351  10484#
CLR   =  000040          1115#  10489
COMPAR   043404          5253   5399   5558   5756   5918   6108   6259   6457   6619   6858   7022   7269   7436
                         7689   7850   8260   8495   8753   9126   10137  11001#
COMREG   042354          2349   2480   2737   2811   3090   3227   3441   3675   3947   4077   4272   4589   4838
                         4960   5232   5379   5541   5737   5900   6089   6241   6439   6601   6838   7003   7249
                         7415   7668   7829   8089   8242   8396   8648   9009   9105   9249   9496   9760   9924
                         10052  10192  10918#
CPHALT   066402          2298   2403   2593   2907   3046   3162   3262   3539   3732   3939   4160   4333   4361
                         4675   5008   5161   5310   5472   5666   5931   6018   6172   6367   6532   6745   6935
                         7177   7344   7591   7762   8009   8173   8466   8724   8937   8966   9061   9190   9298
                         9427   9554   9691   9862   13311#
CR    =  000015          67#    11304  11314
CRLF  =  000200          68#    11275  11314
CSF   =  000002          1244#
CSR   =  000010          1246#
CSR1     002362          1383#  2673   11698  13571  13584  13587  13604  13606
CSR2     002360          1385#  2652   2962   11690  13584  13597  13606
CSR3     002352          1382#
CSR   =  100000          1185#
CSTK  =  000100          1258#
CSTLEAR  002430          1414#  3322
CSTSP =  177570          74#    273    1551
CTRL  =  000040          1157#
CTRL2 =  000002          1153#
CTRLS =  000001          1152#
DAT      071476          333    345    356    13614#
DATIO    071530          410    424    438    454    13626#
DATSAV   071537          573    1050   10862  10874  10886  13630#
DATSET   071547          594    613    629    646    703    741    765    801    837    946    957    988    1007
```

DCRCIA.P11 CROSS REFERENCE TABLE

```
                    13634#
DH4     071505        369   13617#
DH5     071514        374     464     476     490     504     516     524     531     541     550     561     584     625
                      623     639     657     669     682     694     719     732     754     794     813     826     850
                      893     905     971     981     999    1019    1038   13620#
DH51    071554        780   13637#
DH6     071521        389   13622#
DH60    071562        867   13640#
DH61    071570        976   13642#
DH62    071574        885   13644#
DH65    071600        917   13646#
DH66    071605        935   13648#
DH7     071535        397   13624#
DH77    071615      10229   13652#
DH1     066642        326     339     350   13344#
DH10    067440        402     416     430     446   13413#
DH20    067611        565    1043    1055    1067    1079   13433#
DH30    070016        589     611     627     644     705     739     763     799     835     944     955     986    1035
                    13455#
DH4     067021        363   13354#
DH5     067162        374     459     471     485     499     511     522     529     539     548     559     582     603
                      621     637     655     666     680     692     717     730     752     792     811     823     848
                      891     903     966     979     997    1017    1033   13381#
DH51    070132        774   13470#
DH6     067301        385   13395#
DH60    070271        861   13487#
DH61    070426        871   13504#
DH62    070521        880   13515#
DH65    070614        911   13526#
DH66    070737        927   13540#
DH7     067400        394   13406#
DH77    071035       1023   13552#
DH10  = 000004       1154#
DISPLA  001142        273#    1551#   1553#   11173#  11617#
DISPRE  000174        175#    1559
DLAT  = 100000       1125#
DLS64 = 000020       1156#
DMD   = 000001       1183#    2524    2543    2614    2802    3890    3909    4112    4131   13331
DOOG    017262       3875    3899#
DPB   = 000400       1160#    2260    2457    2677    4043    4235    4522    4553    4803    4923    5205    5352    5514
                      5710    5873    6062    6214    6411    6574    6788    6976    7221    7388    7635    7804    8055
                      8215    8359    8611    9222    9469    9734   10527
DR7   = 000200       1159#    2262    2467    2691    4017    4217    4559    4777    4997    5025    5048    7929    8341
                      9554    8593    9836    9326    9587   10527
DST     002366       1390#   11738
DSWR  = 177570         73#     272    1550
DS1     002404       1397#    2677    2681   10334   10346   11706   13584   13587   13604
DS7     002406       1398#    2046#   3417   11810
DTE   = 010000       1182#
DTSY  = 001000       1195#
DT1     071170        332     344     355   13571#
DT10    071266        409     423     437     453   13584#
DT20    071306        572    1049    1061    1073    1085   13587#
DT30    071330        593     612     628     645     706     740     764     800     836     945     956     987    1036
```

```
                    13590#
DT4       071210      368   13574#
DT5       071230      378     463   475   489   503   515   523   530   540   549   562   583   604
                      622     638   656   667   681   693   718   731   753   793   812   824   845
                      892     904   970   980   998  1018  1037  13577#
DTS:      071344      779   13593#
DTS5      071244      388   13579#
DTS60     071362      966   13597#
DTS6:     071400      975   13600#
DTS6S     071416      884   13602#
DTS65     071424      916   13604#
DTS66     071440      934   13606#
DTS7      071258      396   13581#
DTS7      071460     1028   13610#
DZA    =  004000     1145#    2117  2273  2448  2679  4026  4226  4503  4786  4306  5136  5243  5505
                     5701    5864  6053  6205  6402  6565  6779  6967  7212  7379  7626  7795  8046
                     8206    8350  8602  9213  9460  9725 10518
DZC    =  000400     1176#
DZT    =  004000     1270#    3659  3831  4956  5144  5292  5293  5294  5454  5455  5456  5649  5813
                     5814    5815  6001  6154  6155  6156  6350  6514  6515  6516  6726  6915  6916
                     6917    7161  7326  7327  7328  7574  7744  7745  7746  7992  8154  8155  8165
                     8457    8458  8459  8715  8716  8717  8949  9043  9044  9045  9161  9162  9163
                     9410    9411  9412  9670  9671  9672  9854  9970 10105 10106 10107
EC:       002412     1400#   11794
EC2       002414     1401#   11802
EMTVEC=   000030      162#    1535*  1536*  1695*  1716*
EM1       050570      324   12046#
EM10      051322      400   12108#
EM100     066010     1032   13265#
EM101     066072     1041   13274#
EM102     066154     1053   13283#
EM103     066233     1065   13291#
EM104     066317     1077   13300#
EM11      051371      413   12115#
EM12      051476      427   12127#
EM13      051604      441   12139#
EM14      051726      457   12153#
EM15      052001      467   12161#
EM16      052142      479   12179#
EM17      052270      493   12195#
EM2       050617      336   12050#
EM20      052425      507   12213#
EM21      052571      520   12231#
EM22      052644      527   12239#
EM23      052717      534   12247#
EM24      053062      544   12265#
EM25      053225      553   12283#
EM26      053407      564   12303#
EM27      053477      576   12313#
EM3       050706      348   12060#
EM30      053715      587   12340#
EM31      053775      597   12349#
EM32      054212      608   12375#
EM33      054321      615   12387#
```

```
EM34    054523          526   12411#
EM35    054547          632   12415#
EM36    054765          642   12440#
EM37    055042          649   12448#
EM4     050762          359   12069#
EM40    055257          660   12474#
EM41    055466          671   12498#
EM42    055747          685   12529#
EM43    056225          697   12560#
EM44    056557          710   12600#
EM45    057061          722   12633#
EM46    057355          735   12666#
EM47    057470          744   12679#
EM5     051100          372   12082#
EM50    057752          757   12711#
EM51    060171          768   12737#
EM52    060436          784   12756#
EM53    060713          797   12796#
EM54    061001          804   12806#
EM55    061246          816   12835#
EM56    061534          828   12867#
EM57    061744          840   12891#
EM6     051152          382   12089#
EM60    062236          853   12925#
EM61    062647          870   12972#
EM62    062730          879   12981#
EM63    063011          889   12990#
EM64    063065          897   12998#
EM65    063304          910   13023#
EM66    063326          921   13026#
EM67    063520          937   13047#
EM7     051240          392   12098#
EM70    064010          948   13080#
EM71    064216          960   13105#
EM72    064457        13134#
EM73    064707          984   13162#
EM74    064761          973     991   13173#
EM75    065225         1002   13199#
EM76    065330         1010   13211#
EM77    065565         1022   13239#
ERFLGS  004734         1484#  11613#
ERR     = 040000       1166#   2712    2781    5029   10042   10334   10346
ERRVEC= 000004          155#   1548    1549#   1560#   1687#   1695#   1719#  11135   11136#  11139#  11141#
ERWORD  004602         1443#  11014#  11020#  13590
ER1     002364         1389#  11714   13584   13587   13606
ER2     002370         1391#  11722   13587   13606
ER3     002376         1394#  11730   13587   13606
EXT1    = 000001       1224#
EXT10   = 000010       1227#
EXT2    = 000002       1225#
EXT20   = 000020       1228#
EXT4    = 000004       1226#
EXT40   = 000040       1229#
FE4     = 000200       1250#
```

```
FER    = 000020          1174#
FILL     042312         10781#
FILLRE   041270          3340    3343    3386    3389    3392    3395    3398    3401    3408    3413    3422    3425    3645
                         3648    3817    3820    4056    4059    4066    4069    4247    4954    5218    5221    5224    5365
                         5368    5371    5527    5530    5533    5723    5726    5729    5886    5989    5992    6075    6078
                         6081    6227    6230    6233    6424    6427    6430    6587    6590    6593    6824    6827    6830
                         6989    6992    6995    7234    7237    7240    7401    7404    7407    7648    7651    7654    7657
                         7660    7816    7819    7822    8068    8071    8075    9078    8081    8228    8231    8234    8372
                         8624    8994    8997    9000    9090    9093    9096    9235    9238    9241    9482    9485    9488
                         9747    9750    9753   10024   10432#
FINACC   004664          1458#  10589#
FINALA   004662          1457#  10589#
FIRST    004736          1485#   1573    1610#
FISH     017316          3896    3919#
FLHEAD   041212          5095    5117    7093    7100    7121    7134    7298    7496    7513    7534    7547    7717    7343
                         7962    8122    8427    8685    8848    8915   10375#
FMT22  = 010000          1271#   3657    3829    4956    5143    5144    5145    5292    5293    5294    5454    5455    5456
                         5648    5649    5650    5813    5814    5815    6000    6001    6002    6154    6155    6156    6349
                         6350    6351    6514    6515    6516    6725    6726    6727    6915    6916    6917    7160    7161
                         7162    7326    7327    7328    7573    7574    7575    7744    7745    7746    7991    7992    7993
                         8154    8155    8156    8457    8458    8459    8715    8716    8717    8948    8949    8950    9043
                         9044    9045    9161    9162    9163    9410    9411    9412    9670    9671    9672    9853    9854
                         9855    9969    9970    9971   10105   10106   10107
FUTABL   002422          1410#
GNS    = ****** U          174    1581    1585    1590    1594    1598    1603    1607    1662    1704    1708    1767    1794
                         1798    1802    1806    1853    1859    1865    1937    1941    1973    1985    1991    2024    2030
                         2036    2066    2070    2074    2080    2217    2223    2249    2255    2638    2644    2664    2670
                         2914    2918    2932    2938    2942    2969    2975    3011    3015    3512    3523    4346    4350
                         4541    4607    4613    6809    6817    9902    9908   10169   10175   10232   10239   10886   10892
                        10899   10903   10907   10914   10918   11044   11050   11062   11068   11074   11078   11084   11090
                        11094   11104   11672   11680   11698   11696   11704   11712   11720   11728   11736   11744   11752
                        11760   11768   11776   11784   11792   11800   11808   11916   11978   11979   11980   11981   11982
                        11984   11986   11987   11988   11989   11990
GO     = 000001          1139#   2322    2427    2835    2993    3072    3187    3323    3615    3628    3787    3800    4007
                         4026    4207    4226    4368    4483    4503    4569    4629    4766    4786    4886    4906    5017
                         5040    5174    5196    5323    5343    5485    5505    5679    5701    5844    5864    6031    6053
                         6185    6205    6380    6402    6545    6565    6757    6779    6947    6967    7190    7212    7357
                         7379    7604    7626    7775    7795    7919    8022    8046    8186    8206    8329    8350    8479
                         8543    8581    8602    8737    8825    8979    9074    9193    9213    9317    9440    9460    9579
                         9705    9725    9875    9993   10121   10339   10613
GRV    = 000010          1155#
GTSWR  = 104405          1617   11984#
HCE    = 000200          1177#
HCI    = 002000          1269#   3661    3833    4956    5145    5294    5456    5650    5815    6002    6156    6351    6516
                         6727    6917    7162    7329    7575    7746    7993    8156    8459    8717    8950    9045    9163
                         9412    9672    9855    9971   10107
HCRC   = 000400          1178#
HT     = 000011            65#  11273   11314
HT.    = 003323          2330#   2335    2437#   2442    2844#   2849    3002#   3007    3080#   3085    3195#   3200    3331#
                         3336    3636#   3641    3808#   3813    4015#   4020    4215#   4220    4376#   4381    4639#   4643
                         4775#   4780    4895#   4900    5023#   5028    5046#   5051    5185#   5190    5332#   5337    5494#
                         5499    5690#   5695    5853#   5858    6042#   6047    6194#   6199    6391#   6396    6554#   6559
                         6768#   6773    6956#   6961    7201#   7206    7368#   7373    7615#   7620    7794#   7799    7937#
                         7932    8035#   8040    8195#   8200    8339#   8344    8496#   8491    8552#   8557    8591#   8596
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
                        8744*  8749   8834*  8839   8986*  8991   9081*  9085   9202*  9207   9324*  9329   9449*
                        9454   9585*  9590   9714*  9719   9883*  9888   10000* 10005  10128* 10133
IAE    = 002000         1180*
IE     = 000100         1140*  2106   2145   3628   3800   4007   4026   4030   4031   4207   4226   4507   4509
                        4766   4786   4790   4791   4886   4906   4910   4911   5017   5040   5174   5196   5323
                        5343   5485   5505   5679   5701   5844   5864   6031   6053   6195   6205   6380   6402
                        6545   6565   6757   6779   6947   6967   7190   7212   7357   7379   7604   7626   7775
                        7795   7919   8022   8046   8186   8206   8329   8350   8479   8543   8581   8602   8737
                        8825   8979   9074   9193   9213   9317   9440   9460   9578   9705   9725   9875   9993
                       10121  10639  10651  10719  10732
ILF    = 000001         1170*
ILLEGL   002464         1428*  13593
ILR    = 000002         1171*
INUNIT   004752         1495*  2923*  2981
IOTVEC= 000020           160*  1533*  1534*
IR     = 000100         1116*  2232   3358   3379   10012  10013
IXE    = 004000         1254*
KIPAR0= 172340           234*
KIPAR1= 172342           235*
KIPAR2= 172344           236*
KIPAR3= 172346           237*
KIPAR4= 172350           238*
KIPAR5= 172352           239*
KIPAR6= 172354           240*
KIPAR7= 172356           241*
KIPDR0= 172300           223*
KIPDR1= 172302           224*
KIPDR2= 172304           225*
KIPDR3= 172306           226*
KIPDR4= 172310           227*
KIPDR5= 172312           228*
KIPDR6= 172314           229*
KIPDR7= 172316           230*
LA       002420         1403*  11778
LBT    = 002000         1162*  10526
LF     = 000012           66*  11308  11314
LT.    = 000105         2330*   2335   2437*  2442   2844*  2849   3002*  3007   3090*  3095   3195*  3200   3331*
                        3336    3636*  3641   3808*  3813   4015*  4020   4215*  4220   4376*  4381   4638*  4643
                        4775*   4780   4895*  4900   5023*  5028   5046*  5051   5185*  5190   5332*  5337   5494*
                        5499    5690*  5695   5853*  5858   6042*  6047   6194*  6199   6391*  6396   6554*  6559
                        6768*   6773   6956*  6961   7201*  7206   7368*  7373   7615*  7620   7784*  7789   7927*
                        7932    8035*  8040   8195*  8200   8339*  8344   8486*  8491   8552*  8557   8591*  8595
                        8744*   8749   8834*  8839   8986*  8991   9081*  9086   9202*  9207   9324*  9329   9449*
                        9454    9585*  9590   9714*  9719   9883*  9888   10000* 10005  10129* 10133
MAKECY   041036         2525    2544   3891   3910   4113   4132   10323*
MCLK   = 000002         1190*
MCPE   = 020000         1146*  2271   2703   2772
MHS    = 001000         1252*
MINX   = 000004         1191*
MMVEC  = 000250          212*
MCL    = 010000         1164*  2062   2076   2226   2258   2457   3518   3530   4043   4235   4342   4524   4525
                        4555   4556   4803   4923   5205   5352   5514   5710   5873   6062   6214   6411   6574
                        6788   6976   7221   7388   7635   7804   8055   8215   8359   9611   9222   9469   9734
                       10527
```

```
MPE    = 000400          1118*
MR       002402          1396*  11770
MRD    = 000020          1193*
MSE    = 000020          1247*
MSTCK  = 000010          1192*
MWR    = 000040          1194*
MXF    = 001000          1119*   3362
NED    = 010000          1122*   2652    2721    2788    2962
NEM    = 004000          1121*   6804
NHS    = 002000          1253*
NOPERA   002422          1411*   3050    3071    3165    3186
NOPUSH   004724          1478*   1513*   1517*   1520*   2204    2599    2887    3499    4313    9815
NOUNIT   004720          1474*   1823*   1873*   1880    1947*  10256*
NUNIT    004722          1476*   1880*   1881*   3206    3382    4027    4504    4571    4787    4907
OCYL   = 100000          1301*
OF       002372          1392*  11762
OFREV  = 000200          1268*
OFSET    041352          5011   10479*
OFSETC   002454          1424*   4745    4765    4785    5016   10480
OFSTVL   004610          1450*   4677*   4678*   4715    4980*   4981    4983*   4984
OF100  = 000004          1263*
OF200  = 000010          1264*
OF25   = 000001          1261*
OF400  = 000020          1265*
OF50   = 000002          1262*
OF800  = 000040          1266*
OPERSE   042512         10881*  11356   11439
OPI    = 020000          1183*
OR     = 000200          1117*  10012   10013
PAR    = 000010          1173*   3434
PC     =.000007            86*   1572*   1761*   1909*   2061*   2096*   2135*   2195*   2243*   2292*   2296*   2358*   2391*
                          2397*   2401*   2489*   2523*   2542*   2587*   2591*   2651*   2676*   2747*   2755*   2820*   2901*
                          2905*   2952*   2961*   3040*   3044*   3099*   3111*   3160*   3236*   3256*   3260*   3450*   3492*
                          3533*   3537*   3684*   3726*   3730*   3856*   3889*   3908*   3933*   3937*   4086*   4111*   4130*
                          4154*   4158*   4281*   4327*   4331*   4355*   4359*   4384*   4598*   4618*   4669*   4673*   4693*
                          4687*   4847*   4859*   4968*   4976*   5002*   5006*   5034*   5087*   5159*   5241*   5262*   5269*
                          5308*   5388*   5408*   5429*   5470*   5550*   5567*   5603*   5664*   5746*   5765*   5773*   5829*
                          5909*   5927*   5965*   6016*   6098*   6117*   6124*   6170*   6250*   6268*   6304*   6365*   6447*
                          6466*   6474*   6530*   6610*   6628*   6671*   6743*   6848*   6868*   6875*   6933*   7013*   7032*
                          7076*   7175*   7257*   7278*   7285*   7342*   7425*   7446*   7489*   7589*   7677*   7698*   7705*
                          7760*   7839*   7860*   7888*   7935*   8007*   8097*   8108*   8171*   8251*   8269*   8304*   8405*
                          8464*   8504*   8528*   8657*   8672*   8722*   8762*   8801*   8805*   8841*   8964*   9017*   9029*
                          9059*   9115*   9136*   9146*   9178*   9258*   9292*   9296*   9332*   9425*   9504*   9541*   9552*
                          9593*   9689*   9770*   9829*   9860*   9933*   9948*  10061*  10075*  10201*  10285*  10288*  10298*
                         10303   10333*  10345*  10351*  10492*  10533*  10538*  10830*  10840*  10951*  11021*  11052*  11282*
                         11289*  11296*  11310*  11312*  11340*  11354*  11415*  11460*  11627*  11872*  12028
PCJSR    041422          10325*  10326*  10504*  10506*  10507*  10512*  10513*  13600   13602   13610
PCLBUF   002346           1362*  10612*
PCLCSR   002344           1361*  10586*  10613*
PCLCTR   002350           1363*  10587
PGE    = 002000           1120*
PIP    = 020000           1165*   4043    4235    4522    4553    4803    4923    8359    9611
PIRQ   = 177772             72*
PIRQVE = 000240            166*
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
PKACK   002460          1426*  2300   2321   2405   2426   2447   2834   2992   4367   4628
PLU   = 020000          1256*
PRE   = 000020          1296*
PRITEM  046760          1653*  2372*  2948*  3698*  4616*  10210* 10255* 10883* 11659* 11821* 11838  11842*
PROG  = 001000          1161*
PRO   = 000000            89*
PR1   = 000040            90*
PR2   = 000100            91*
PR3   = 000140            92*
PR4   = 000200            93*
PR5   = 000240            94*
PR6   = 000300            95*
PR7   = 000340            96*
PS    = 177776            69*    70   1565*  2105*  2144*  2199*  10228* 10892*
PSEL  = 002000          1144*  6731   6757   6779   6921   6947   6967   9678   9705   9725
PSU   = 000001          1293*
PSW   = 177776            70*
PUTREG  043300          2651   2676   2961   10333  10345  10830  10938*
PWRVEC= 000024           161*  1539*  1540*  11996* 11997* 12006* 12012* 12024* 12025*
RDCHR = 104407         11514  11987*
RDLIN = 104410         11561  11988*
RDOCT = 104411          1664  10909  10920  11053  11070  11989*
RDY   = 000200          1141*  2106   2117   2145   2273   2448   2697   3082   3197   3333   3810   4026   4226
                        4494   4503   4786   4906   5187   5334   5496   5692   5855   6044   6196   6393   6556
                        6770   6958   7203   7370   7617   7786   8037   8197   8350   8488   8602   8746   8988
                        9083   9204   9451   9716   9885   10002  10130  10518
READAT  002446          1421*  5457   5484   5504   5816   5843   5863   6157   6184   6204   6517   6544   6564
                        6918   6946   6966   10108  10120
READIN  002462          1427*  3591   3614   3627   3763   3786   3799
RECALI  002426          1413*  3987   4006   4025   4187   4206   4225   5039   7918   8542
REFOR   002450          1422*  5295   5322   5342   7329   7355   7378   7747   7774   7794   8157   8185   8205
                        8460   8478   8718   8736   9046   9073
REGADR  004600          1442*  1738*  2284*  2454*  2463*  4040*  4049*  4232*  4241*  4517*  4533*  4800*  4809*
                        4920*  4929*  5202*  5211*  5349*  5358*  5511*  5520*  5707*  5716*  5870*  5879*  6059*
                        6068*  6211*  6220*  6408*  6417*  6571*  6580*  6785*  6794*  6973*  6982*  7218*  7227*
                        7385*  7394*  7632*  7641*  7801*  7810*  8052*  8061*  8212*  8221*  8356*  8365*  8609*
                        8617*  9219*  9228*  9466*  9475*  9731*  9740*  10356* 10839* 13577  13579  13593  13597
                       13610
REGSAV  046614         11612*
REGSA1  046622          1685  11614*
REINTO  003534          1436*  5118   5125   5254   5274   5288   5370   5401   5450   5532   5560   5626   5631
                        5757   5778   5783   5809   5891   5920   5983   6109   6129   6150   6232   6261   6327
                        6332   6458   6479   6484   6510   6592   6621   6703   6708   6859   6880   6885   6911
                        6994   7024   7122   7129   7135   7142   7270   7290   7322   7406   7438   7535   7542
                        7548   7555   7690   7710   7740   7821   7852   8113   8150   8233   8262   8420   8453
                        8497   8677   8711   8755   8931   9039   9095   9128   10082  10101  10139
RELEAS  002432          1415*
RESVEC= 000010           156*
RETCL   002456          1425*  4864   4885   4905
RHAS    002316          1343*  1732   1769   3590*  3762*  4851*
RHBA    002274          1331*  3344   5222   5369   5511   5727   5890   6079   6231   6428   6591   6928   6993
                        7238   7405   7652   7820   8012   8232   8998   9094   9239   9486   9751   10010  10971*
RHBAE   002340          1355*
RHCA    002312          1341*  3399   3646   3818   765    8079   10328* 10452* 10470* 10969*
```

RHCC	002334	1350*	2282	2284	2286	3313	4057	4248	7659	8076	8373	8625	10352	10356
		10588	10944											
RHCS1	002300	1336*	2300*	2324*	2405*	2429*	2693	2764	2[illegible]*	2995*	3050*	3074*	3081	3165*
		3189*	3196	3325*	3332	3365	3430	3591*	3617*	3630*	3763*	3789*	3802*	3809
		3987*	4009*	4067	4187*	4209*	4263	4370*	4461*	[illegible]95*	4493	4565	4575	4631*
		4745*	4768*	4818	4864*	4888*	4938	5019*	5042*	51[illegible]	5196	5325*	5333	5487*
		5495	5681*	5691	5846*	5854	6033*	6043	6187*	6195	[illegible]382*	6392	6547*	6555
		6731*	6759*	6769	6921*	6949*	6957	7192*	7202	7359*	[illegible]	7606*	7616	7777*
		7785	7921*	8024*	8036	8188*	8196	8331*	8388	8481*	848[illegible]	8545*	8583*	8640
		8739*	8745	8827*	8981*	8987	9076*	9082	9195*	9203	9319*	[illegible]42*	9450	9580*
		9678*	9707*	9715	9877*	9884	9995*	10001	10017	10123*	10129	103[illegible]	10339*	10454*
		10471*	10480*	10484	10639	10651	10719	10732	10977*	11046	11080			
RHCS2	002276	1332*	1770	1829*	1955*	2717	2784	3354	3375	6804	10011	10485	10[illegible]*	
RHCS3	002342	1356*												
RHDB	002270	1329*	1690	11054										
RHDST	002304	1338*	2650	2960	3390	3649	3821	5225	5372	5534	5730	5893	6082	623[illegible]
		6431	6594	6831	6996	7241	7408	7661	7823	8082	8235	9001	9097	9242
		9489	9754	10025	10329*	10453*	10969*							
RHDS1	002322	1345*	2331	2339	2438	2469	2706	2775	2845	3003	3215	3414	3637	3664
		3836	4016	4070	4216	4256	4340	4377	4554	4639	4776	4825	4896	4945
		5024	5029	5047	7928	8340	8381	8553	8592	8633	8835	9325	9586	9915
		10036	10182	10486										
RHDT	002324	1346*	1830	1832	1836	1838	1841	1843	1861	1956	1958	1965	1967	1993
		2000	2002	2005	2007	2010	2012	2044	2046					
RHEC1	002330	1348*	3423											
RHEC2	002332	1349*	3426											
RHER1	002302	1337*	1773	3387	10029	10487								
RHER2	002306	1339*	3393											
RHER3	002314	1342*	2724	2791	3402									
RHLA	002336	1351*	10589											
RHMR	002320	1344*	2524*	2543*	2614*	2798	3409	3890*	3909*	4112*	4131*	10331*		
RHOF	002310	1340*	3396	3653	3825	4060	4955	10479*	10975*					
RHSN	002326	1347*	1987	2043	2045									
RHWC	002272	1330*	2307	2412	2619	3057	3115	3172	3268	3341	3541	3598	3734	3770
		3942	3994	4163	4194	4390	4468	4692	4751	4871	5153	5219	5302	5366
		5464	5528	5658	5724	5823	5887	6010	6076	6164	6228	6359	6425	6524
		6588	6737	6825	6927	6990	7169	7235	7336	7402	7583	7649	7754	7817
		8001	8069	8165	8229	8314	8566	8958	8995	9053	9091	9172	9236	9419
		9483	9683	9748	9894	9978	10009	10160	10437	10756	10839	10942	10944	10970*
RH70	004750	1493*	1624*	3204	3349	3611	3783	6678	9548					
RH70CK	006146	1622*												
RMR =	000004	1172*												
RPTRP1	011370	2102	2116*											
RPTRP2	011476	2141	2155*											
RPVEC	002266	1094*	1567	2101	2140	2313*	2418*	3063*	3178*	3314*	3604*	3776*	3999*	4199*
		4474*	4757*	4877*	5165*	5314*	5476*	5670*	5835*	6022*	6176*	6371*	6536*	6749*
		6939*	7181*	7348*	7595*	7766*	7911*	8013*	8177*	8320*	8470*	8534*	8572*	8728*
		8816*	8970*	9065*	9184*	9308*	9431*	9569*	9695*	9866*	9984*	10112*	11064	11071*
		11086	11107*											
RPVECT	044352	1568	11101*	11107										
RPOS	004746	1492*	1964*	1970*	2505	2576	2869							
RPO6	004744	1491*	1954*	1961*	2503	2574	2867	3872	4095					
RP4VEC	004606	1445*	1652*	2313	2418	3063	3178	3314	3604	3776	3999	4199	4474	4757
		4877	5165	5314	5476	5670	5835	6022	6176	6371	6536	6749	6939	7181

DZRJIA.P11 CROSS REFERENCE TABLE

RUN 043340

R0 =:.000000

7348 7595 7766 7911 8013 8177 8320 8470 8534 8572 8728 8816 8970
9065 9184 9308 9431 9569 9695 9866 9984 10112
5133 5282 5445 5639 5804 5991 6145 6340 6505 6716 6906 7150 7316
7563 7734 7981 8144 8447 8705 8938 9033 9151 9401 9661 9844 9960
10096 10968#
77# 1567* 1568* 1569* 1689* 1692* 1772* 1776* 1816* 1619* 1824* 1876* 1917*
1918 1928* 1929 1931 1946 1949* 1955 1976 2101* 2102* 2103* 2140* 2141*
2142* 2270* 2271* 2273 2306* 2338* 2349* 2411* 2433* 2450 2453 2468* 2480*
2525* 2544* 2618* 2650* 2692* 2705* 2716* 2723* 2737* 2763* 2774* 2783* 2790*
2797* 2811* 2841* 2960* 2999* 3056* 3090* 3115* 3116* 3118* 3120* 3121* 3123*
3125* 3127* 3129* 3131* 3133* 3137 3141* 3144* 3153* 3154 3157* 3171* 3215*
3227* 3268* 3269* 3271* 3273* 3274* 3276* 3278* 3280* 3282* 3284* 3286* 3290
3294* 3297* 3306* 3307 3310* 3340* 3343* 3353* 3364* 3374* 3386* 3389* 3392*
3395* 3398* 3401* 3408* 3413* 3422* 3425* 3429* 3441* 3541* 3546* 3548* 3550*
3551* 3553* 3555* 3557* 3559* 3561* 3563* 3567 3571* 3574* 3583* 3584 3587*
3597* 3621* 3645* 3648* 3652* 3663* 3675* 3734* 3739* 3741* 3743* 3744* 3746*
3748* 3750* 3752* 3754* 3756* 3758* 3759* 3769* 3793* 3817* 3820* 3824* 3835*
3947* 3891* 3910* 3942* 3943* 3945* 3947* 3948* 3950* 3952* 3954* 3956* 3958*
3960* 3964 3968* 3971* 3980* 3981 3984* 3993* 4012* 4029 4036 4039 4056*
4059* 4066* 4069* 4077* 4113* 4132* 4163* 4165* 4167* 4169* 4170* 4172* 4174*
4176* 4178* 4180* 4182* 4184* 4185* 4193* 4212* 4228 4231 4247* 4255* 4262*
4272* 4390* 4395* 4399* 4403* 4406* 4410* 4414* 4418* 4422* 4426* 4430* 4436
4440* 4443* 4452* 4453 4458* 4467* 4489* 4506 4513 4516 4564* 4574* 4589*
4635* 4692* 4694* 4697* 4700* 4702* 4705* 4708* 4711* 4715* 4717* 4719* 4723
4727* 4730* 4739* 4740 4743* 4750* 4771* 4789 4796 4799 4817* 4824* 4838*
4870* 4891* 4909 4916 4919 4937* 4944* 4954* 4960* 5011* 5095* 5105* 5117*
5124* 5133* 5152* 5179* 5198 5201 5218* 5221* 5224* 5232* 5253* 5273* 5282*
5301* 5328* 5345 5348 5365* 5368* 5371* 5379* 5399* 5436* 5445* 5463* 5490*
5507 5510 5527* 5530* 5533* 5541* 5558* 5610* 5615* 5625* 5630* 5639* 5657-
5684* 5703 5706 5723* 5726* 5729* 5737* 5756* 5777* 5782* 5790* 5795* 5804*
5822* 5849* 5866 5869 5886* 5889* 5892* 5900* 5918* 5972* 5982* 5991* 6009*
6036* 6055 6058 6075* 6078* 6081* 6089* 6108* 6128* 6136* 6145* 6163* 6190*
6207 6210 6227* 6230* 6233* 6241* 6259* 6311* 6316* 6326* 6331* 6340* 6358*
6385* 6404 6407 6424* 6427* 6430* 6438* 6457* 6478* 6483* 6491* 6496* 6505*
6523* 6550* 6567 6570 6587* 6590* 6593* 6601* 6619* 6687* 6692* 6702* 6707*
6716* 6736* 6762* 6781 6784 6824* 6827* 6830* 6838* 6859* 6879* 6884* 6892*
6897* 6906* 6926* 6952* 6969 6972 6989* 6992* 6995* 7003* 7022* 7083* 7092*
7100* 7109* 7121* 7128* 7134* 7141* 7150* 7168* 7195* 7214 7217 7234* 7237*
7240* 7248* 7269* 7289* 7298* 7307* 7316* 7335* 7362* 7381 7384 7401* 7404*
7407* 7415* 7436* 7496* 7505* 7513* 7522* 7534* 7541* 7547* 7554* 7563* 7582*
7609* 7628 7631 7648* 7651* 7654* 7657* 7660* 7668* 7689* 7709* 7717* 7726*
7734* 7753* 7780* 7797 7800 7816* 7819* 7822* 7829* 7850* 7924* 7943* 7953*
7962* 7972* 7981* 8000* 8027* 8048 8051 8068* 8071* 8075* 8078* 8081* 8089*
8112* 8122* 8130* 8135* 8144* 8164* 8191* 8209 8211 8228* 8231* 8234* 8242*
8260* 8309* 8313* 8334* 8352 8355 8372* 8380* 8387* 8396* 8419* 8427* 8434*
8439* 8447* 8495* 8548* 8561* 8565* 8586* 8604 8607 9624* 8632* 9639* 8648*
8676* 8685* 8692* 8697* 8705* 8753* 8813* 8930* 8848* 8859* 9864* 9869* 8874*
8879* 8888* 8892 8893* 8900* 8901 8902* 9909* 8915* 9922* 8930* 8938* 8957*
8994* 8997* 9000* 9008* 9033* 9052* 9090* 9093* 9096* 9105* 9126* 9151* 9171*
9198* 9215 9218 9235* 9238* 9241* 9249* 9305* 9339* 9344* 9349* 9354* 9359*
9368* 9372 9373* 9380* 9381 9382* 9388* 9393* 9401* 9418* 9445* 9462 9465
9482* 9485* 9488* 9496* 9566* 9600* 9605* 9610* 9615* 9620* 9629* 9633 9634*
9641* 9642 9643* 9648* 9653* 9661* 9682* 9710* 9727 9730 9747* 9750* 9753*
9760* 9836* 9844* 9893* 9914* 9924* 9952* 9960* 9977* 10015* 10024* 10029* 10035*

```
                 10352*  10381*  10388*  10396*  10137*  10159*  10181*  10192*  10258*  10260   10295*  10298   10325
                 10327   10361*  10378   10379   10383   10388*  10408   10409   10410   10417*  10435   10436   10441*
                 10452   10453   10456*  10470   10472*  10479   10481*  10555   10556   10557   10564*  10614   10618*
                 10619   10621   10622   10623   10624   10625   10677*  10697   10699*  10700   10702   10703   10704
                 10705   10741*  10754   10755   10757   10759   10761   10766*  10786   10787   10788   10789   10793*
                 10924   10925   10826   10827   10829*  10856*  10939   10942*  10945   10950*  10968   10969   10970
                 10971   10973   10975   10977   10978*  11007   11008   11009   11010   11011*  11035*  11054*  11056*
                 11057*  11190   11200*  11204   11220   11221   11234*  11265   11266*  11267   11270*  11558   11562*
                 11563   11566   11586*  11589*  11824   11825*  11826*  11833*  11834*  11835*  11836*  11837*  11838
                 11840   11942   11843   11948   11854   11856*  11857   11871*  11962   11963*  11964   11965*  11966*
                 11967*  11968*  11998   12023*
R1      =:000001   78*   1690*   1691    1697    1732*   1733*   1734    1736    1738    1759*   1779    1782    1825
                 2106*   2117    2145*   2270    2276    2433    2454    2841    2999    3621    3793    4012    4040
                 4212    4232    4489    4517    4635    4771    4800    4891    4920    5173    5202    5328    5349
                 5490    5511    5684    5707    5849    5970    6036    6059    6190    6211    6385    6408    6550
                 6571    6762    6785    6952    6973    7195    7218    7362    7385    7609    7632    7780    7801
                 7924    8027    8052    8191    8212    8334    8356    8548    8586    8608    8830    8897    8890*
                 8992*   8901*   8904*   9198    9219    9367    9370*   9372*   9391*   9384*   9445    9466    9628
                 9631*   9633*   9642*   9645*   9710    9731    10259*  10260   10263   10376   10378*  10383*  10387*
                 10405   10408*  10411*  10416*  10433   10435*  10437*  10438*  10440*  10484*  10491*  10516   10521
                 10552   10555*  10556   10563*  10615   10623*  10657*  10658   10667*  10668   10676*  10752   10754*
                 10756*  10759*  10761*  10765*  10782   10786*  10791*  10798*  10819   10824*  10835   10837   10842
                 10855*  10940   10943*  10945*  10949*  11002   11007*  11015   11018   11022   11034*  11055*  11058*
                 11191   11204*  11205   11209   11233*  11559   11564*  11572*  11574*  11576*  11579*  11582   11585*
                 11853   11854*  11859   11863   11865   11870*  11999   12022*
R2      =:000002   79*   1691*   1770*   1771*   1775*   2113*   2121*   2152*   2158*   2233    2236    3139    3142*
                 3148*   3151*   3292    3295*   3301*   3304*   3569    3572*   3578*   3581*   3966    3969*   3975*
                 3978*   4438    4441*   4447*   4450*   4725    4728*   4734*   4737*   10377   10379*  10384*  10396*
                 10406   10409*  10412*  10415*  10434   10436*  10438   10439*  10485*  10489*  10490*  10553   10556*
                 10558*  10562*  10616   10624*  10657   10666*  10667   10675*  10753   10755*  10762*  10764*  10783
                 10787*  10793*  10797*  10820   10825*  10835   10838   10842   10854*  10941   10944*  10946*  10949*
                 11003   11008*  11015   11019   11022   11033*  11192   11203*  11207*  11210   11217*  11218*  11219
                 11224*  11232*  11560   11565*  11573*  11575*  11577*  11583   11584*  12000   12021*
R3      =:000003   80*   1917*   1818*   1821*   1869*   2062    2076    2226    2228    2258    2260    2262    2264
                 2266    2434    2463    2842    3000    3518    3530    3622    3794    4013    4049    4213    4241
                 4490    4533    4636    4772    4809    4892    4929    5180    5211    5329    5358    5491    5520
                 5665    5716    5850    5879    6037    6068    6191    6220    6396    6417    6551    6580    6763
                 6794    6953    6982    7196    7227    7363    7394    7610    7641    7781    7910    7925    8028
                 8061    8192    8221    8335    8365    8549    8587    8617    8831    9199    9228    9446    9475
                 9711    9740    10407   10410*  10411   10414*  10486*  10508   10514   10529   10554   10557*  10559*
                 10561*  10617   10628*  10631*  10633*  10636*  10674*  10698   10708*  10711*  10713*  10716*  10740*
                 10784   10788*  10791   10792*  10796*  10821   10826*  10847*  10953*  11004   11009*  11012   11020
                 11027*  11032*  11193   11201*  11202*  11216*  11219*  11228*  11229*  11231*  11510   11511*  11512
                 11515*  11516   11520   11522   11524*  11526*  11910   11919*  11925*  11926*  11929*  11934*  11935*
                 11936   11945*  12001   12020*
R4      =:000004   81*   1773*   1774*   3138    3146*   3152*   3291    3299*   3305*   3568    3576*   3582*   3965
                 3973*   3979*   4437    4445*   4451*   4724    4732*   4738*   10487*  10795   10789*  10792   10795*
                 10822   10827*  10840   10852*  11005   11012*  11013*  11014   11031*  11911   11913*  11914*  11915*
                 11916   11917*  11931   11933*  11941*  11944*  12002   12019*
R5      =:000005   82*   1922*   1829    1855    1869    1871    1875*   2434*   2459    2462    2942*   3000*   3143*
                 3149*   3296*   3302*   3573*   3579*   3622*   3794*   3970*   3976*   4013*   4045    4048    4213*
                 4237    4240    4442*   4448*   4490*   4523    4529    4532    4636*   4729*   4735*   4772*   4905
                 4909    4892*   4925    4928    5180*   5207    5210    5329*   5354    5357    5491*   5515    5519
                 5665*   5712    5715    5850*   5875    5878    6037*   6064    6057    6191*   6215    6217    6396*
```

```
                    6413    6416    6551*   6576    6579    6763*   6790    6793    6953*   6978    6981    7196*   7223
                    7226    7363*   7390    7393    7510*   7637    7640    7781*   7906    7909    7925*   8029*   8057
                    8060    8192*   8217    8220    8335*   8361    8364    8549*   8587*   8613    8616    8831*   8889*
                    8994*   9199*   9224    9227    9369*   9374*   9446*   9471    9474    9630*   9635*   9711*   9736
                    9739   10324   10327*  10328   10353   10355   10360*  10823   10832*  10934*  10939   10951*  11006
                   11010*  11021   11030*  11194   11196*  11198*  11205*  11209*  11224   11230*  11912   11918*  11920*
                   11922*  11923*  11924*  11925   11943*  12003   12018*
R0      =:000000    93*     95    1527*   1528*   1529
R1      =:000001    84*     86
SAVER   041536    2206    2411    2619    3056    3171    3597    3769    3993    4193    4467    4750    4870    5152
                   5301    5463    5667    5822    6009    6163    6359    6523    6736    6926    7168    7335    7582
                   7753    8000    8164    8313    8565    8957    9052    9171    9418    9682    9893    9977   10159
                  10551*
SAVERE  034612    1453*   2309    2351    2413    2492    2620    2730*   2731*   2739    2804*   2813    3058    3092
                   3173    3208*   3212*   3229    3313*   3346*   3347*   3384*   3407*   3417*   3419*   3443    3539
                   3677    3771    3849    3995    4063*   4079    4195    4251*   4274    4469    4560*   4582*   4591
                   4752    4831*   4840    4872    4951*   4962    5154    5234    5303    5381    5465    5543    5659
                   5739    5824    5902    6011    6091    6165    6243    6360    6440    6525    6603    6738    6840
                   6928    7005    7170    7250    7337    7417    7594    7670    7755    7831    8002    8091    8166
                   8244    8315    9376*   8398    8567    8628*   8650    8959    9010    9054    9107    9173    9251
                   9420    9498    9684    9762    9895    9926    9979   10009*  10010*  10013*  10014*  10044*  10054
                  10161   10194   10438*  10759*  10761*  10831
SC      = 100000    1148*   2271    2699    2768    3208    3369    3384    4033    4267    4510    4579    4793    4822
                   4913    4942    8392    8644   10021
SC1     = 000100    1230*
SC10    = 001000    1233*
SC200   = 000200    1231*
SC20    = 000020    1234*
SC4     = 000040    1232*
SEECOM  002452    1423*   8328    8349    8580    8601    8824    9316    9577   10330   10471
SEEKCY  041336    8309    9561    8813    9305    9566   10470*
SELECT  004726    1480*   1514*   1516*   1519*   1658    1884    1921   10244
SELFST  007730    1788    1884*
SERC4   002434    1416*  10454
SK1     = 040000    1300*
SP      = 000040    1399*   2045*   3419   11818
SP1     = 000100    1576    1610*
SP      =:000006      85*   1531*   1548*   1556*   1560    1665*   1666    1667    1684*   1696    1711*   1729*   1955*
                   1861*   1871*   1906*   1981*   1987*   1993*   2094*   2116    2133*   2155    2192*   2219*   2231*
                   2232*   2233    2251*   2321*   2322*   2324    2394*   2426*   2427*   2429    2447*   2449*   2449
                   2450    2457*   2458    2459    2520*   2539*   2584*   2640*   2666*   2834*   2835*   2837    2999*
                   2934*   2971*   2992*   2993*   2995    3037*   3071*   3072*   3074    3137*   3139*   3139*   3151
                   3152    3153    3186*   3187*   3189    3253*   3290*   3291*   3292*   3304    3305    3306    3322*
                   3323*   3325    3404*   3406*   3407    3489*   3515*   3526*   3566*   3569*   3569*   3581    3592
                   3683    3614*   3615*   3617    3627*   3628*   3630    3723*   3786*   3787*   3799    3799*   3800*
                   3802    3886*   3905*   3930*   3964*   3965*   3966*   3978    3979    3980    4006*   4007*   4009
                   4025*   4026*   4031*   4032*   4033*   4035    4036    4043*   4044    4045    4109*   4127*   4151*
                   4206*   4207*   4209    4225*   4226*   4227    4228    4235*   4236    4237    4324*   4340*   4341*
                   4342    4367*   4368*   4370    4436*   4437*   4438*   4450    4451    4452    4492*   4493*   4495
                   4502*   4503*   4508*   4509*   4510*   4512    4513    4522*   4525*   4526*   4528    4529    4533*
                   4556*   4557*   4559*   4560    4609*   4628*   4629*   4631    4666*   4723*   4724*   4725*   4737
                   4738    4739    4765*   4766*   4768    4785*   4786*   4791*   4792*   4793*   4795    4796    4803*
                   4804    4805    4885*   4886*   4888    4905*   4906*   4911*   4913*   4913*   4915    4916    4923*
                   4924    4925    4939*   5016*   5017*   5019    5039*   5040*   5042    5094*   5173*   5174*   5176
```

```
                    5195*  5196*  5197   5198   5205*  5206   5207   5322*  5323*  5325   5342*  5343*  5344
                    5345   5352*  5353   5354   5426*  5484*  5485*  5487   5504*  5505*  5506   5507   5514*
                    5515   5516   5600*  5678*  5679*  5681   5700*  5701*  5702   5703   5710*  5711   5712
                    5843*  5844*  5846   5863*  5864*  5865   5866   5873*  5874   5875   5962*  6030*  6031*
                    6033   6052*  6053*  6054   6055   6062*  6063   6064   6184*  6185*  6187   6204*  6205*
                    6206   6207   6214*  6215   6216   6301*  6379*  6380*  6382   6401*  6402*  6403   6404
                    6411*  6412   6413   6544*  6545*  6547   6564*  6565*  6566   6567   6574*  6575   6576
                    6669*  6756*  6757*  6759   6778*  6779*  6780   6781   6788*  6789   6790   6946*  6947*
                    6949   6966*  6967*  6968   6969   6976*  6977   6978   7073*  7189*  7190*  7192   7211*
                    7212*  7213   7214   7221*  7222   7223   7356*  7357*  7359   7378*  7379*  7380   7381
                    7388*  7389   7390   7486*  7603*  7604*  7606   7625*  7626*  7627   7628   7635*  7636
                    7637   7774*  7775*  7777   7794*  7795*  7796   7797   7804*  7805   7806   7885*  7918*
                    7919*  7921   8021*  8022*  8024   8045*  8046*  8047   8048   8055*  8056   8057   8185*
                    8186*  8188   8205*  8206*  8207   8208   8215*  8216   8217   8328*  8329*  8331   8349*
                    8350*  8351   8352   8359*  8360   8361   8478*  8479*  8481   8525*  8542*  8543*  8545
                    8580*  8581*  8583   8601*  8602*  8603   8604   8611*  8612   8613   8736*  8737*  8739
                    8798*  8824*  8825*  8827   8887*  8904   8978*  8979*  8981   9073*  9074*  9076   9192*
                    9193*  9195   9212*  9213*  9214   9215   9222*  9223   9224   9289*  9316*  9317*  9319
                    9367*  9384   9439*  9440*  9442   9459*  9460*  9461   9462   9469*  9470   9471   9538*
                    9577*  9579*  9580   9628*  9645   9704*  9705*  9707   9724*  9725*  9726   9727   9734*
                    9735   9736   9826*  9874*  9875*  9877   9904*  9992*  9993*  9995   10011*  10012*  10014
                   10120*  10121*  10123  10171*  10234*  10240*  10250*  10292*  10324*  10360  10376*  10377*  10386
                   10387   10405*  10406*  10407*  10414   10415   10416   10433*  10434*  10439  10440   10506   10508*
                   10509*  10512   10514*  10516*  10517*  10518   10526*  10527   10537*  10552*  10553*  10554*  10561
                   10562   10563   10614*  10615*  10616*  10617*  10618   10625*  10674   10675   10676   10677   10697*
                   10698*  10699   10705*  10740   10741   10752*  10753*  10764   10765   10782*  10783*  10784*  10795*
                   10795   10796   10797   10798   10819*  10820*  10821*  10822*  10823*  10843*  10844*  10845   10951
                   10852   10853   10854   10855   10889*  10894*  10910*  10911   10921   10922*  10939*  10940*  10941*
                   10948   10949   10950   10972*  10973*  10974   11002*  11003*  11004*  11005*  11006*  11023*  11024*
                   11025   11030   11031   11032   11033   11034   11046*  11057   11064*  11071   11080*  11086*  11135*
                   11138   11140   11141   11169   11170   11174*  11190*  11191*  11192*  11193*  11194*  11195*  11196
                   11199*  11212   11214*  11216   11226   11228   11230   11231   11232   11233   11234   11236*  11237*
                   11265*  11266   11267*  11269   11270   11271*  11273   11275   11277   11283   11285*  11297*  11295*
                   11299   11303   11304   11308   11349*  11350*  11351   11355   11357   11366   11368   11371   11374*
                   11375*  11376   11381   11383   11385*  11386   11402*  11403*  11404   11414   11421*  11424*  11425*
                   11429*  11430*  11432   11435*  11442   11445*  11449   11451   11453   11454*  11461   11463   11465*
                   11466   11468*  11469*  11470*  11471*  11472*  11487*  11488*  11489*  11490*  11491*  11497*  11510*
                   11515   11526   11527*  11528*  11529*  11556*  11557*  11558*  11559*  11560*  11562   11566*  11568
                   11570   11578*  11579   11581   11582*  11584   11585   11586   11588   11622   11636*  11639*  11661*
                   11662*  11663   11674*  11682*  11690*  11698*  11706*  11714*  11722*  11730*  11738*  11746*  11754*
                   11762*  11770*  11778*  11786*  11794*  11802*  11810*  11818*  11824*  11829*  11853*  11859*  11863*
                   11870   11871   11902*  11903   11904   11905*  11910*  11911*  11912*  11919   11943   11944   11945
                   11946*  11947*  11962*  11963   11999*  11999*  12000*  12001*  12002*  12003*  12004*  12005   12013*
                   12017   12018   12019   12020   12021   12022   12023   12029*
SRCH    54:316     10452#
SRC     = 177572     216#
SR1     = 177574     217#
SR2     = 177576     218#
SR3     = 172516     219#
STACK   = 001000      60#  1531   1684   1729   1906   2094   2133   2192   2394   2520   2539   2584   2999
                    3037   3253   3489   3723   3886   3905   3930   4108   4127   4151   4324   4666   4999
                    5084   5426   5600   5962   6301   6669   7073   7496   7885   8525   8799   9289   9533
                    9826
START   005022      1515   1518   1523*
```

```
STKLMT= 177774          71#
        032450          7899*   7941#   8282*   9295
        033052          7906*   8129#   8290*
        033076          7907*   9142#   8291*
        033232          9247    8250#
        033336          8248    8258#
        033354          8264    8268#
        033360          8265    8282#
        033446          9293    9295#
        033452          8294    8302#
        033702          9401    9404#
        033706          8402    9418#
        032470          7899*   7952#   9283*
        040256          10166#
                        9170#
                        8499    8503#
                        8500    8508#
        032502          7900*   7960#   8284*
        032522          7901*   7971#   8285*
        032534          7902*   7979#   8296*
        032742          7903*   7904*   8074#   9287*   9288*
                        8094    8096#
                        8095    8106#
        033032          7905*   8120#   8289*
        001140          272#    1529    1550*   1552    1558*   1615    1622    1699    1763    1786    2979    10943   11023
                        11130   11144   11146   11152   11159   11359   11398   11453*  11618   11625   11630   11634   11661
                        12004   12017*
SWREG   000176          176#    1558    1615    11359   11398   11421
SW00  = 000001          124#
SW01  = 000002          114#    124
SW02  = 000004          113#    123
SW03  = 000010          112#    122
SW04  = 000020          111#    121
SW05  = 000040          110#    120
SW06  = 000100          109#    119
SW07  = 000200          108#    118     11653
SW08  = 000400          107#    117     10845   11025
SW09  = 001000          106#    116
SW10  = 002000          105#    115
SW11  = 004000          123#
SW12  = 010000          104#
SW13  = 020000          103#
SW14  = 040000          102#    1622
SW15  = 100000          101#    1698    1763    1786
                        100#
                        99#
SW2     000004          122#
SW3     000010          121#
SW4     000020          120#
SW5     000040          119#    2879
SW6     000100          118#
SW7     000200          117#
SW8     000400          116#
SW9     001000          115#
```

CROSS REFERENCE TABLE

```
TBITVE= 000014          157*
TDF   = 000040         1249*
TESTAD  042510        10980*
TIMCNT  042064         2108    2147   10628   10633   10694*  10709   10713
TIMER1  041E66        10596*
TIMER2  041600         1552   10588*
TKVEC = 000060          164*  11336*  11337*
TMP0    004754         1502*
TMP5    004756         1503*   4997*   5052*
TMP4    00476C         1504*   4029*   4030*   4032   4506*  4507*   4509   4523*  4524*  4526   4554*  4555*  4557
                       4789*   4790*   4792   4909*  4910*  4912
TN    = 000050         1906*   1907   2192*  2193   2394*  2395   2584*  2585   2898*  2899   3037*  3038   3253*
                       3254   3489*  3490   3723*  3724   3930*  3931   4151*  4152   4324*  4325   4666*  4667
                       4999*  5000   5084*  5085   5426*  5427   5600*  5601   5962*  5963   6301*  6302   6668*
                       6669   7073*  7074   7486*  7487   7985*  7986   8525*  9526   8798*  8799   9289*  9290
                       9538*  9539   9826*  9827
TOTALA  004742         1489*  1779*  1781*  3141   3294   3404   3571   3968   4440   4727
TRVEC = 000064          165*
TRAPVE= 000034          163*  1537*  1538*  1566*
TRE   = 040000         1147*  2271   2701   2770   10023
TRK1  = 004000         1235*
TRK1C = 040000         1238*
TRK2  = 010000         1236*
TRK2C = 100000         1239*
TRK4  = 020000         1237*
TRTVEC= 000014          158*
TSTNM   004604         1444*  1683*  1730*  1759*  1907*  2060*  2093*  2132*  2193*  2395*  2521*  2540*  2595*
                       2899*  3038*  3254*  3490*  3724*  3887*  3906*  3931*  4109*  4128*  4152*  4325*  4667*
                       5000*  5085*  5427*  5601*  5963*  6302*  6669*  7074*  7487*  7986*  8526*  8799*  9290*
                       9539*  9827*  10888   13571   13574   13577   13579   13591   13594   13597   13590   13593   13597
                      13600   13602   13604   13606   13610
TST1    006312         1659    1671    1681*  10304
TST10   011506          [illegible]  2191*
TST11   012266          [illegible]  2386*
TST12   012550         25.9*
TST13   012610         2510    2538*
TST14   012644         2531    2570*
TST15   013656         2580    2608    2686    2363*
TST16   014664         2507    2873    2881    2896    3036*
TST17   015322         3252*
TST2    006564         1727*
TST20   016110         3488*
TST21   016654         3508    3722*
TST22   017222         3885*
TST23   017262         3904*
TST24   017316         3929*
TST25   017776         4107*
TST26   020036         4126*
TST27   020072         4150*
TST3    006640         1735    1757*
TST30   020440         4308*
TST31   021754         4322    4664*
TST32   023016         4987    4996*
TST33   023170         5053    5093*
```

```
TST34   024056          5425*
TST35   024372          5599*
TST36   025302          5961*
TST37   026142          6300*
TST4    000444          1885    1903*  10253   10264
TST40   027056          6666*
TST41   030000          6680    7072*
TST42   031230          7485*
TST43   032262          7884*
TST44   034076          8524*
TST45   034650          8797*
TST46   035756          9288*
TST47   036472          9537*
TST5    010732          2059*
TST50   037240          9550    9810*
TST51   040464          9824    10226*
TST6    011276          2063    2091*
TST7    011436          2114    2118    2130*
TT      = 000020        1249*
TYPDS   = 104404        1856    1872    1982    2220    2252    2641    2667    2935    2972    3516    3527    4610    9905
                        10172   10235   10241   10251   10293   11864   11982*
TYPE    = 104400        1579    1583    1588    1592    1596    1601    1605    1660    1702    1706    1765    1792    1796
                        1800    1804    1851    1857    1963    1870    1935    1939    1977    1983    1989    2022    2029
                        2034    2064    2068    2072    2079    2215    2221    2247    2253    2298    2403    2593    2636
                        2642    2662    2668    2907    2912    2916    2930    2936    2940    2967    2973    3009    3013
                        3046    3162    3262    3510    3517    3521    3528    3539    3732    3939    4160    4333    4344
                        4348    4361    4539    4605    4611    4675    5008    5161    5310    5472    5666    5831    6019
                        6172    6367    6532    6745    6807    6911    6812    6915    6919    6920    6935    7177    7344
                        7591    7762    8009    8173    8466    8724    8807    8966    9061    9180    9298    9427    9554
                        9691    9862    9900    9906    10167   10173   10230   10236   10249   10252   10291   10294   10884
                        10890   10896   10897   10901   10905   10912   10916   11042   11048   11060   11066   11072   11076
                        11082   11098   11092   11102   11235   11278   11353   11365   11419   11420   11423   11434   11444
                        11455   11474   11518   11521   11525   11590   11592   11620   11628   11679   11678   11686   11694
                        11702   11710   11718   11726   11734   11742   11750   11758   11766   11774   11782   11790   11798
                        11806   11814   11823   11845   11847   11850   11852   11867   11937   11978*  12026
TYPERR  047722          11666   11822*
TYPCC   = 104401        1713    1862    1988    1994    10889   10895   11047   11065   11081   11087   11106   11422   11575
                        11683   11691   11699   11707   11715   11723   11731   11739   11747   11755   11763   11771   11779
                        11787   11795   11803   11811   11819   11831   11860   11979*
TYPCN   = 104403        11981*
TYPOS   = 104402        11980*
UBUSB   004732          1482*   6683*   6806*   9558
UNIB    = 000020        1114*
UNIT    004716          1473*   1666*   1672*   1879*   1886*   1915    1946*   1949    1981    2219    2231    2251    2643
                        2666    2923    2934    2949*   2950*   2971    2981    3347    3515    3526    4609    9904    10171
                        10231   10256   10263*  10490   10972
UNITS   004676          1472*   1817    1921    1879    1928    10259
UNITSL  004730          1481*   1667*   1672    1896
UNLOAD  002424          1412*   4461    4482    4502
UNS     = 040000        1184*
UPE     = 020000        1123*
US1     = 000001        1110*
US2     = 000002        1111*
US4     = 000004        1112*
```

```
.WR     = 000010  1295#
.WF     = 000002  1294#
.WU30   = 010000  1255#
.WV     = 000100  1158#  2264  2332  2343  2439  2457  2473  2714  2846  3004  3638  3668  3840
                  4043  4235  4342  4378  4524  4525  4555  4556  4640  4903  4923  5205  5352
                  5514  5710  5873  6062  6214  6411  6574  6788  6976  7221  7388  7635  7804
                  8055  8215  8359  8611  9222  9469  9734 10509 10527

WAITBT  041622  10609# 10622* 10629  10634  10703* 10709  10714  13571  13574
WAITPC  041616  10607# 10619* 10620* 10700* 10701* 13571  13574
WAITRE  041620  10608# 10621* 10629  10634  10638  10650  10702* 10709  10714  10718  10734  13571  13574
WAITTM  041624  10597* 10610# 10611* 10658  10668  13574
WAIT.P  041626  10611#
WAIT.T  042056  10696# 11990
WAT     = 104412  2330  2437  2844  3002  3080  3195  3331  3636  3808  4015  4215  4376  4492
                  4638  4775  4895  5023  5046  5185  5332  5494  5690  5853  6042  6194  6391
                  6554  6768  6956  7201  7368  7615  7784  7927  8035  8195  8339  8486  8552
                  8591  8744  8834  8986  9081  9202  9324  9449  9585  9714  9883 10000 10129
                 11990#
WC      002354  1383#  2352  2483  2740  2814  3093  3230  3444  3678  3950  4080  4275  4592
                  4841  4963  5235  5382  5544  5740  5903  6092  6244  6441  6604  6841  7006
                  7251  7418  7671  7832  8092  8245  8399  8651  9011  9139  9252  9499  9763
                  9927 10055 10195 10943 11674
.WCE    = 040000  1124#
.WCF    = 000040  1175#
.WCU    = 000001  1243#
.WLE    = 004000  1181# 10033
WRCHDT  002440  1418#  9164  9192  9212
WRCHEK  002436  1417#  9413  9439  9459  9673  9704  9724
WRFROM  002470  1435#  5096  5106  5139  5223  5255  5400  5437  5559  5611  5616  5644  5728
                  5758  5791  5796  5919  5973  5996  6080  6110  6137  6260  6312  6317  6345
                  6429  6459  6492  6497  6620  6698  6693  6721  6829  6860  6893  6898  7023
                  7084  7093  7101  7110  7156  7239  7271  7299  7308  7437  7497  7505  7514
                  7523  7569  7653  7691  7718  7727  7851  7944  7954  7963  7973  7987  8073
                  8123  8131  8136  8261  8428  8435  8440  8496  8686  8693  8698  8754  8849
                  8860  8865  8870  8875  8880  8890  8910  8916  8923  8944  8999  9127  9157
                  9240  9340  9345  9350  9355  9360  9370  9389  9394  9406  9487  9601  9606
                  9611  9616  9621  9631  9649  9654  9666  9752  9837  9949  9953  9965 10099
                 10138
WRICAT  002442  1419#  5651  5678  5700  6003  6030  6052  6352  6379  6401  6729  6756  6779
                  9856  9874  9972  9992
WRIFOR  002444  1420#  5146  5173  5195  7163  7189  7211  7576  7603  7625  7994  8021  8045
                  8951  8978
WRL     = 004000  1163#  9919 10186
WRU     = 000400  1251#
WSU     = 000004  1245#
XE2     007372  1784  1815#
$AUTOB  001134   269#  1619*  11412  11541
$BDADR  001122   264#  2228*  2236*  2266*  2276*  10521*  10529*  13584  13597  13600  13602
$BDDAT  001126   266#  1736*  2286*  2453*  2462*  4039*  4048*  4231*  4240*  4516*  4532*  4799*  4809*
                 4919*  4928*  5201*  5210*  5348*  5357*  5510*  5519*  5706*  5715*  5869*  5879*  6058*
                 6067*  6210*  6219*  6407*  6416*  6570*  6579*  6784*  6793*  6972*  6981*  7217*  7226*
                 7384*  7393*  7631*  7640*  7800*  7809*  8051*  8060*  8211*  8220*  8355*  8364*  8607*
                 8616*  9218*  9227*  9465*  9474*  9730*  9739*  10352*  10353  10638*  10650*  10719*  10734*
                10838*  11019*  13571  13574  13577  13579  13590  13593  13597  13610
```

```
$BELL   001216          298*  11365  11534  11620  11642
$CHARC  045326        11280*  11290* 11297  11306* 11311*
$CkSWR  045640        11398*  11986
$CMTAG  001100          352*   1526   1527   1535   1541   1542
$CM1  = 000006          284*    285*   286*   287*   288*   289*   290*
$CM2  = 000014          284*    285*   286*   287*   288*   289*   290*
$CM3  = 000006          292*    284
$CM4  = 000006          290*    291*   292*   293*   294*   295*   296*
$CNTLC  046411        11353   11434  11534*
$CNTLS  046423        11419   11536*
$CNTLU  046416        11444   11535*
$CRLF   001223          300*   1870   3517   3529   6811   6812   6819   6820  10896  11279  11314  11455  11534
                      11595   11628  11642  11823  11847  11852
$DBLK   045102        11201   11235  11243*
$DORGN  041012        10287   10296  10302*
$DTS_   045072        11204   11239*
$ENDAD  041002          184   10298*
$ENDCT  040750        10289*
$ENDMG  041021        10249   10291  10306*
$ENULL  041016        10252   10294  10305*
$EOP    040714         1714    1809  10257  10279*
$EOPCT  040742        10286*  10290
$ERFLG  001103          255*  11119  11148  11150  11156*  11177  11615*  11642
$ERMAX  001115          261*   1543* 11150  11172*  11177
$ERROR  046612         1535    1716  11609*
$ERRPC  001116          262*  11622* 11623*  11624  11642  11829  13571  13574  13577  13579  13581  13594  13587
                      13590   13593  13597  13600  13602  13604  13606  13610
$ERRTB  001226          317*  11837
$ERRTY  046762        11627   11661*
$ERTTL  001112          259*  10240  10242*  11621*  11642
$ESCAP  001214          297*   1542* 11171*  11637  11639  11642
$FILLC  001156          280*  11283  11314
$FILLS  001155          279*  11314
$GDADR  001120          263*
$GDDAT  001124          265*   1737*  2285*  2449*  2458*  4035*  4044*  4227*  4236*  4512*  4528*  4795*  4804*
                       4915*  4924*  5197*  5206*  5344*  5353*  5506*  5515*  5702*  5711*  5865*  5874*  6054*
                       6063*  6206*  6215*  6403*  6412*  6566*  6575*  6780*  6789*  6968*  6977*  7213*  7222*
                       7380*  7389*  7627*  7636*  7796*  7805*  8047*  8056*  8207*  8216*  8351*  8360*  8603*
                       8612*  9214*  9223*  9461*  9470*  9726*  9735*  10355*  10537*  11019*  13577  13590  13593
                      13597   13610
$GET42  040772        10295*
$GTSWR  045730        11420*  11984
$HD   = 000000           12
$HIOCT  046610        11583*  11594*
$ICNT   001104          256*  11163*  11164  11166*  11176
$ILLUP  050552        11996   12012  12031*
$INTAG  001135          270*  11417*  11436  11456  11541
$ITEMB  001114          260*  11624*  11642  11826
$LF     001224          301*  11314  11525  11534  11595  11642
$LPADR  001106          257*   1544*  1905*  4665*  10911*  10922  11154*  11169*  11174  11176
$LPERR  001110          258*   1545*  9945*  10894  10921*  11154  11170*  11176  11636
$MAIL = ****** U        1562    1615  11169  11267  11630
$MNEW   046441        11423   11539*
$MSWR   046430        11420   11537*
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
$MXCNT  044664          11167  11176*
$NULL   001154            278* 11285  11314
$NWTST= 000001           1675*  1677   1721*  1723   1744*  1746   1890*  1892   2051*  2053   2083*  2085   2124*
                         2126   2164*  2166   2377*  2379   2516*  2535*  2552*  2554   2854*  2856   3028*  3030
                         3245*  3247   3460*  3462   3701*  3703   3882*  3901*  3920*  3922   4104*  4123*  4141*
                         4143   4287*  4289   4645*  4647   4990*  4992   5057*  5059   5414*  5416   5571*  5573
                         5933*  5935   6272*  6274   6635*  6637   7042*  7044   7451*  7453   7866*  7868   8512*
                         8514   8772*  8774   9268*  9270   9515*  9517   9784*  9786  10215* 10217
$OCNT   050332          11909* 11938* 11951*
$OMODE  050334          11904* 11908* 11913  11916* 11927* 11953*
$OVER   044650          11131  11147  11155  11165  11173*
$PASS   001100            253*  2208   2603   2891   3503   4317   9819  10248* 10250  10283* 10284* 10292  10305
                        11161  11177
$POWER  050560          12027  12034*
$PWRAD  050546          12029*
$PWRDN  050406           1539  11996* 12024
$PWRMG  050542          12027*
$PWRUP  050460          12006  12012*
$QUES   001222            299* 11314  11474  11518  11534  11592  11595  11642
$RDCHR  046202          11487* 11987
$RDDEC= ****** L        11990
$RDLIN  046272          11510* 11988
$RDOCT  046452          11556* 11989
$RDSZ = 000011          11503*
$REGAD  001160            282*
$REGO   001162            284*
$REG1   001164            285*
$REG2   001166            286*
$REG3   001170            287*
$REG4   001172            288*
$REG5   001174            289*
$RTNAD  041014          10304*
$R2A  = ****** U        11990
$SAVRE= ****** U        11990
$SAVR6  050556          12005* 12013  12014* 12015* 12033*
$SCOPE  044414           1533  11128*
$SETUP= 000117           1524*  1532   1533   1535   1537   1539   1541   1542   1544   1612  10291  11129  11357
                        11362  11363  11393  11541  11610  11633  11641
$SS1  = 000000           1564*
$STUP = 177777           1524*
$SVLAD  044622          11139  11168*
$SVPC = 000200            182*   187
$SWR  = 167770              2*    12     44     45     46     47     48     49     50     51    296    297    298
                         1541   1542   1544   1545   1682   1729   1758   1904   2060   2092   2131   2192   2397
                         2520   2539   2571   2864   3037   3253   3489   3723   3986   3905   3930   4108   4127
                         4151   4309   4665   4997   5084   5426   5600   5962   6301   6667   7073   7486   7895
                         8525   8798   9289   9538   9811  10227  10276  10282  10297  10303  10305  11120  11121
                        11122  11123  11124  11130  11142  11144  11145  11148  11149  11150  11157  11159  11159
                        11170  11173  11176  11601  11602  11603  11604  11605  11618  11625  11630  11634  11642
                        12030
$SWRMK= 000000            51     52  11124  11125  11146
$TIMES  001212            296*  1541*  1682*  1728*  1758*  1904*  2571*  2864*  6667*  10227*  10292*  11157*  11164
                        11167* 11176
$TKB    001146            275* 11317  11338  11349  11374  11402  11429
```

```
$TKCNT  045332          11318*  11333*  11363   11380*  11494   11496*
$TKINT  045352           1572    1761    2391   11052   11333*  11354   11415
$TKQEN= 045351          11322*  11388   11499
$TKQIN  045334          11319*  11334*  11335   11386*  11387*  11389   11390*
$TKQOU  045336          11320*  11335*  11497   11498*  11499   11501*
$TKQSR  045340          11321*  11334   11390   11501
$TKS    001144            274*  11317   11339*  11370*  11372   11378*  11400   11416*  11426   11438*  11458*
$TKSRV  045422          11336   11349*
$TMP0   001176            290*   1825*   1826*
$TMP1   001200            291*   1697*   2108*   2109*   2147*   2148*   7933*   8292*  13591
$TMP2   001202            292*
$TMP3   001204            293*
$TMP4   001206            294*
$TMP5   001210            295*
$TN   = 000052              2*     12    1659    1671    1675    1682*   1683    1721    1728*   1730    1735    1744    1759*
                         1759    1885    1890    1904*   1906    2051    2060*   2063    2083    2092*   2093    2114    2118
                         2124    2131*   2132    2153    2164    2192*   2212    2377    2387*   2394    2507    2509    2516
                         2520*   2521    2531    2535    2539*   2540    2552    2571*   2580    2584    2607    2686    2854
                         2864*   2873    2881    2895    2898    3028    3037*   3245    3253*   3460    3489*   3507    3701
                         3723*   3882    3886*   3887    3901    3905*   3906    3920    3930*   4104    4108*   4109    4123
                         4127*   4128    4141    4151*   4287    4309*   4321    4324    4645    4665*   4666    4987    4990
                         4997*   4999    5053    5057    5084*   5414    5426*   5571    5600*   5933    5962*   6272    6301*
                         6635    6667*   6668    6680    7042    7073*   7451    7486*   7866    7885*   8512    8525*   8772
                         8798*   9268    9289*   9515    9538*   9550    9784    9811*   9823    9826   10215   10227*
$TPB    001152            277*  11303*  11314
$TPFLG  001157            281*  11261   11314
$TPS    001150            276*  11301   11314
$TRAP   050336           1537   11962*
$TRP  = 000013          11970*  11979*  11980*  11981*  11982*  11983*  11984   11985*  11986   11987*  11988*  11989*  11990*
                        11991*
$TRPAD  050360          11967   11977*
$TSTNM  001102            254*  10243*  10281*  11119   11146   11168*  11173   11177   11617   11642
$TTYIN  046400          11511   11512   11529   11533*
$TYPBN= ****** U        11983
$TYPDS  044666          11189*  11982
$TYPE   045112          11261*  11970   11978
$TYPEC  045262          11282   11289   11296   11301*  11302   11460
$TYPEX  045330          11307   11309   11312*
$TYPOC  050134          11907*  11979
$TYPON  050150          11906   11909*  11981
$TYPOS  050110          11902*  11980
$XTSTR  044426          11133*
$$GET4= 000000          10297*
$DFILL  050333          11903*  11907*  11917   11952*
$40CAT= ****** U        11130   11627
.     = 071624            170*    174*    182     183*    185*    187*    191*    193*    195*    244*    251*    302    1434*
                         1435*   1436*   1453*   1472*   1530    1544    1545    1586*   1595*   1599*   1608*   1795*   1799*
                         1854*   1860*   1866*   1942*   1980*   2025*   2031*   2037*   2071*   2075*   2081*   2224*   2915*
                         2933*   2943*   3012*   3016*   4347*   4614*   6818*   9903*   9909*  10170*  10239*  10305   10309
                        10900*  10915*  11063*  11176   11177   11243*  11314   11317   11321*  11322   11323*  11533*  11534
                        11541   11595   11642   11673*  11681*  11745*  11753*  11761*  11769*  11777*  11785*  11809*  11817*
                        11874*  12008   12032   13569*  13655*
```

DZRJIA.P11 CROSS REFERENCE TABLE

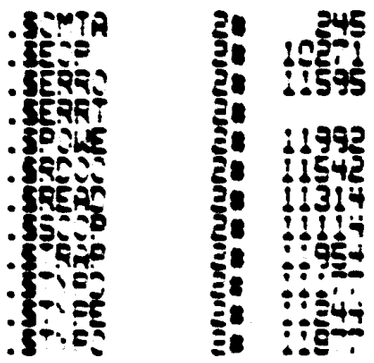
```
ALLREG    40#  11668  11677  11685  11693  11701  11709  11717  11725  11733  11741  11749  11757  11765  11773
          11781  11789  11797  11805  11813
CHANGR    40#   2337   2467   2691   2704   2715   2722   2762   2773   2782   2789   2796   3210   3214   3352
           3363   3372   3428   3651   3662   3823   3834   4062   4250   4254   4261   4563   4573   4580   4816
           4823   4830   4936   4943   4950   8375   8379   8386   8627   8631   8638   9913  10015  10027  10034
          10043  10180
CHECKD    40#   2591   2905   3044   3160   3260   3937   4158   4331   5006   5159   5308   5470   5664   5829
           6016   6170   6365   6530   6743   6933   7175   7342   7589   7760   8007   8171   8464   8722   8805
           8964   9059   9178   9296   9425   9552   9689   9860
CHECKV    40#   2296   2401   3537   3730   4359   4673
CHKCNT    40#   4383   4687   4859
CKCNTV    40#   4618
CLEARA    40#   5105   5124   5273   5436   5610   5615   5625   5630   5777   5782   5790   5795   5972   5982
           6128   6136   6311   6316   6326   6331   6478   6483   6491   6496   6687   6692   6702   6707   6879
           6884   6892   6897   7092   7109   7128   7141   7289   7307   7505   7522   7541   7554   7709   7726
           7952   7971   8112   8129   8135   8418   8434   8439   8676   8692   8697   8859   8864   8869   8874
           8879   8909   8922   8930   9339   9344   9349   9354   9359   9388   9393   9600   9605   9610   9615
           9620   9648   9653   9836   9952  10081  10088
CMPBLK    40#   5251   5397   5556   5754   5916   6106   6257   6455   6617   6856   7020   7267   7434   7687
           7848   8258   8494   8752   9124  10136
CMREGI    40#   2348   2479   2736   2810   3089   3226   3440   3674   3846   4076   4271   4588   4837   4959
           5231   5378   5540   5736   5899   6088   6240   6437   6600   6837   7002   7247   7414   7667   7828
           8088   8241   8395   8647   9007   9104   9248   9495   9759   9923  10051  10191
COMMEN      13    167#
DATACO    40#   5132   5281   5444   5638   5803   5990   6144   6339   6504   6715   6905   7149   7315   7562
           7733   7979   8142   8446   8704   8937   9032   9150   9400   9660   9843   9959  10095
DISREG    40#
DUM       40#   2300   2405   3050   3165   3591   3763   3987   4187   4461   4745   4864
ENDCOM      22    167#
ERROR     61#   1698   1740   2112   2119   2156   2229   2237   2267   2277   2287   2357   2455   2464   2488
           2655   2684   2745   2819   2965   3098   3235   3449   3683   3855   4041   4050   4085   4233   4242
           4280   4518   4534   4597   4801   4810   4846   4921   4930   4967   5031   5203   5212   5240   5261
           5350   5359   5387   5407   5512   5521   5549   5566   5708   5717   5745   5764   5871   5880   5908
           5926   6060   6069   6097   6116   6212   6221   6249   6267   6409   6418   6446   6465   6572   6591
           6609   6627   6786   6795   6846   6866   6974   6993   7011   7030   7219   7228   7256   7277   7386
           7395   7424   7445   7633   7642   7676   7697   7802   7811   7838   7859   8053   8062   8096   8213
           8222   8250   8268   8357   8366   8404   8503   8609   8618   8656   8761   9016   9114   9135   9220
           9229   9257   9467   9476   9503   9732   9741   9768   9932  10060  10145  10200  10336  10348  10357
          10522  10530  10641  10643  10653  10662  10671  10721  10724  10735
ESCAPE    167#
FIHEAD    40#   5094   5116   7082   7099   7120   7133   7297   7495   7512   7533   7546   7716   7941   7960
           8120   8426   8684   8847   8914
FILLBL    40#
FLSVRE    40#   3340   3343   3386   3389   3392   3395   3398   3401   3408   3412   3422   3425   3645   3648
           3817   3820   4056   4059   4066   4069   4247   4954   5218   5221   5224   5365   5368   5371   5527
           5530   5533   5723   5726   5729   5886   5889   5892   6075   6078   6081   6227   6230   6233   6424
           6427   6430   6587   6590   6593   6824   6827   6830   6989   6992   6995   7234   7237   7240   7401
           7404   7407   7648   7651   7654   7657   7660   7816   7819   7822   8068   9071   8074   8078   8091
           8228   8231   8234   8372   8624   8994   8997   9000   9090   9093   9096   9235   9238   9241   9482
           9485   9488   9747   9750   9753  10024
GETPRI    167#
GETSWR      2#   167#  1612
GOO       40#   2320   2425   2833   2991   3070   3185   3321   3613   3625   3785   3797   4005   4205   4366
           4481   4627   4764   4884   5015   5038   5172   5321   5483   5677   5842   6029   6183   6379   6543
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
        6755  6945  7188  7355  7602  7773  7917  8020  8184  8327  8477  8541  8579  8735  8823
        8977  9072  9191  9315  9438  9576  9703  9873  9991 10119
LOAD      40* 3116  3118  3120  3121  3123  3125  3127  3129  3131  3133  3135  3157  3269  3271
        3273  3274  3276  3278  3280  3282  3284  3286  3288  3310  3546  3548  3550  3551  3553
        3555  3557  3559  3561  3563  3565  3587  3739  3741  3743  3744  3746  3748  3750  3752
        3754  3756  3758  3759  3943  3945  3947  3948  3950  3952  3954  3956  3958  3960  3962
        3984  4165  4167  4169  4170  4172  4174  4176  4178  4180  4182  4184  4185  4395  4399
        4403  4406  4410  4414  4418  4422  4426  4430  4434  4458  4694  4697  4700  4702  4705
        4708  4711  4717  4719  4721  4743
MAKECL    40* 2515  2534  3881  3899  4103  4121
MANUAL    40* 2201  2596  2884  3496  4310  9812
MSG     1674* 1677  1720* 1723  1743* 1746  1889* 1892  2050* 2053  2082* 2085  2123* 2126  2164*
        2166  2376* 2379  2551* 2554  2853* 2856  3027* 3030  3245* 3247  3459* 3462  3699* 3703
        3918* 3922  4139* 4143  4287* 4289  4645* 4647  4990* 4992  5057* 5059  5413* 5416  5570*
        5573  5933* 5935  6272* 6274  6635* 6637  7041* 7044  7450* 7453  7865* 7868  8511* 8514
        8772* 8774  9267* 9270  9514* 9517  9783* 9786 10215* 10217
MULT      167*
NEWTST    167* 1675  1721  1744  1890  2051  2083  2124  2154  2377  2516  2535  2552  2854  3028
        3245  3460  3701  3882  3901  3919  4104  4123  4140  4287  4645  4990  5057  5414  5571
        5933  6272  6635  7042  7451  7866  8512  8772  9268  9515  9784 10215
OFFST     40* 5011
POP       167* 3153  3306  3583  3980  4452  4739  8904  9384  9645 10359 10386 10414 10439 10561
       10673 10739 10764 10795 10850 10948 11029 11230 11584 12017 12018
PUSH      167* 3137  3290  3567  3964  4436  4723  8887  9367  9628 10323 10375 10404 10432 10551
       10614 10696 10751 10781 10818 10938 11001 11189 11558 11998 12004
REPORT    167*
RFORGC    40*
RHCLEA    40* 1908  2095  2134  2194  2241  2290  2396  2586  2753  2900  2351  3039  3109  3255
        3491  3532  3725  3932  4153  4326  4354  4668  4681  4974  5001  5033  5086  5267  5428
        5602  5771  5964  6122  6303  6472  6670  6873  7075  7283  7488  7703  7887  7934  8106
        8302  8527  8670  8800  8840  9026  9144  9291  9331  9540  9592  9828  9947 10073
RH70CK    40* 6675  9545
SAVE      40* 11611
SAVTST    40* 1683  1730  1759  2060  2093  2132  2521  2540  3887  3906  4109  4128
SCH       40*
SCOPE     62* 1681  1727  1757  1903  2059  2091  2130  2191  2386  2519  2538  2570  2863  3036
        3252  3488  3722  3885  3904  3929  4107  4126  4150  4308  4664  4996  5032  5083  5425
        5599  5961  6300  6666  7072  7485  7884  8524  8797  9288  9537  9810 10226 10280
SEEKCO    40* 8308  8560  8812  9304  9564
SETPRI    167* 11490
SETTRA 11970* 11979 11980 11981 11982 11984 11986 11987 11988 11989 11990
SETUP     167* 1525
SKIP      40*  167* 1659  1671  1735  1885  2063  2114  2118  2153  4987  5053
SLASH     167*
SPACE     167*
SREGIS    40* 2306  2411  2618  3056  3171  3597  3769  3993  4193  4467  4750  4870  5152  5301
        5463  5657  5822  6009  6163  6358  6523  6736  6926  7168  7335  7592  7753  8000  8164
        8313  8565  8957  9052  9170  9418  9682  9893  9977 10158
STARS       32    38  167*  180   243   247   302  1089  1091  1131  1133  1374  1379  1626  1648
        1675  1680  1721  1726  1744  1756  1835  1845  1890  1902  1953  1972  1999  2038  2051
        2058  2083  2090  2124  2129  2164  2190  2377  2385  2502  2512  2516  2518  2535  2537
        2552  2569  2573  2582  2854  2862  2866  2875  3028  3035  3245  3251  3460  3487  3701
        3721  3871  3877  3882  3884  3901  3903  3920  3928  4094  4100  4104  4106  4123  4125
        4141  4149  4287  4307  4645  4663  4857  4990  4995  5057  5082  5414  5424  5571  5598
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
            5933    5960    6272    6299    6635    6665    7042    7071    7451    7484    7866    7883    8512    8523    8772
            8796    9268    9287    9515    9536    9784    9809   10213   10214   10215   10225   10273   11099   11109   11116
           11179   11246   11316   11393   11408   11479   11503   11544   11597   11879   11956   11994   12010
STARTT       40#    1906    2192    2394    2584    2898    3037    3253    3489    3723    3930    4151    4324    4666    4998
            5024    5426    5600    5962    6301    6668    7073    7486    7885    8525    8798    9289    9538    9826
SWRSU       167#   1546#
TJUMP        40#    2212    2507    2509    2531    2580    2607    2686    2873    2881    2895    3507    4321    6680    9550
            9823
TRMTRP    11970#
TSCLR2       40#
TSCLR5       40#
TTSTNO       40#    1907    2193    2395    2585    2899    3038    3254    3490    3724    3931    4152    4325    4667    5000
            5085    5427    5601    5963    6302    6669    7074    7487    7886    8526    8799    9290    9539    9827
TYPBIN      167#
TYPDEC      167#   10292
TYPNAM      167#
TYPNUM      167#
TYPOCS      167#
TYPOCT      167#    1987    1993   11421
TYPTXT      167#    1578    1583    1588    1592    1596    1601    1605    1660    1702    1706    1765    1791    1796    1800
            1804    1851    1857    1863    1934    1939    1977    1983    1989    2021    2027    2033    2064    2068    2072
            2078    2215    2221    2247    2253    2636    2642    2662    2668    2912    2916    2929    2936    2940    2967
            2973    3009    3013    3510    3521    4344    4348    4539    4604    4611    6807    6814    9900    9906   10166
           10173   10230   10236   10884   10890   10897   10901   10905   10912   10916   11041   11048   11060   11066   11072
           11076   11082   11088   11092   11101   11670   11678   11686   11694   11702   11710   11718   11726   11734   11742
           11750   11758   11766   11774   11782   11790   11798   11806   11814
VECSET       40#    2313    2418    3063    3178    3314    3604    3776    3999    4199    4474    4757    4877    5165    5314
            5476    5670    5835    6022    6176    6371    6536    6749    6939    7181    7348    7595    7766    7911    8013
            8177    8320    8470    8534    8572    8728    8816    8970    9065    9184    9308    9431    9569    9695    9866
            9984   10112
WT           40#    2329    2436    2843    3001    3079    3194    3330    3634    3806    4014    4214    4375    4637    4774
            4894    5022    5045    5184    5331    5493    5689    5852    6041    6193    6390    6553    6767    6955    7200
            7367    7614    7783    7926    8034    8194    8338    8485    8551    8590    8743    8833    8985    9080    9201
            9323    9448    9584    9713    9882    9999   10127
WTT          40#    2335    2442    2849    3007    3085    3200    3336    3641    3813    4020    4220    4381    4643    4780
            4900    5028    5051    5190    5337    5499    5695    5858    6047    6199    6396    6559    6773    6961    7206
            7373    7620    7789    7932    8040    8200    8344    8491    8557    8596    8749    8839    8991    9086    9207
            9329    9454    9590    9719    9888   10005   10133
$$CMRE      245#    284     285     286     287     288     289
$$CMTM      245#    290     291     292     293     294     295
$$ESCA      167#
$$NEWT      167#    1675    1721    1744    1890    2051    2083    2124    2164    2377    2516    2535    2552    2854    3029
            3245    3460    3701    3882    3901    3920    4104    4123    4141    4287    4645    4990    5057    5414    5571
            5933    6272    6635    7042    7451    7866    8512    8772    9268    9515    9784   10215
$$SET     11970#   11979   11980   11981   11982   11984   11986   11987   11988   11989   11990
$$SKIP      167#
.EQUAT        2#      57
.HEADE        2#
.KT11         2#     208
.SETUP        2#    1523
.SWRHI        2#      40
.SWRLO        2#      52#     53
.$ACTI        2#     178
.$CATC        2#     168
```



CROSS REFERENCE TABLE

```
ADC      4983    9293    9295    9290   10537   10666   10667   10792   10834   10910   11209   11271   11435   11445   11454
        11579   11837   11905   11915
ASL      9902    9392    9643   11469   11469   11470   11572   11574   11576   11834   11835   11836   11966
ASLB    11214
         1623    3145    3299    3575    3972    4444    4731    8903    9383    9644   11215
BEQ      1523    1658    1671    1735    1783    1797    1831    1833    1837    1839    1842    1844    1885    1957    1953
         1966    1968    2001    2003    2006    2008    2011    2013    2077    2118    2227    2235    2259    2261    2263
         2665    2275    2283    2452    2461    2678    2680    2682    2963    3205    3531    3612    3784    3873    4028
         4038    4047    4096    4230    4239    4343    4505    4515    4531    4788    4798    4807    4908    4918    4927
         4982    5030    5053    5200    5209    5347    5356    5509    5518    5705    5714    5868    5877    6057    6066
         6209    6215    6406    6415    6569    6578    6679    6783    6792    6805    6971    6980    7216    7225    7383
         7392    7630    7639    7799    7808    8050    8059    8210    8219    8354    8363    8606    8615    9217    9226
         9464    9473    9549    9559    9729    9738   10245   10257   10261   10296   10335   10347   10354   10520   10529
        10640   10652   10720   10733   10758   10836   10846   11016   11026   11145   11147   11149   11153   11162   11274
        11308   11360   11413   11452   11467   11495   11567   11616   11619   11635   11638   11844   11849   11855   11865
        11932
        11165
        10297   11223   11384   11464   11569   11939
        11151
        10661   10670
         1665    2271    2804    2950    3206    3212    3384    3406    4030    4031    4341    4507    4508    4524    4525
         4555    4556    4559    4582    4790    4791    4910    4911   10012   10013   10284   10517   10526   10761   10944
        11224   11256   11350   11375   11385   11403   11430   11465   11578   11662   11929
BIS      2232    2322    2427    2448    2614    2730    2835    2993    3072    3120    3187    3273    3323    3347    3550
         3615    3629    3743    3787    3800    3947    4007    4026    4032    4033    4063    4169    4207    4226    4251
         4368    4403    4483    4503    4509    4510    4526    4557    4629    4700    4756    4786    4792    4793    4831
         4886    4906    4912    4913    4951    5017    5040    5174    5196    5323    5343    5495    5505    5679    5701
         5944    5864    6031    6053    6195    6205    6380    6402    6545    6565    6731    6757    6779    6921    6947
         7067    7190    7212    7357    7379    7604    7626    7775    7795    7919    8022    8046    8186    8206    8329
         8350    8376    8479    8543    8581    8602    8628    8737    8825    8979    9074    9193    9213    9317    9440
         9460    9578    9678    9705    9725    9875    9993   10014   10044   10121   10339   10479   10509   10759   10973
        11057   11217   11219   11472   11934   11935
        11826
         1622    1639    1763    1796    2062    2076    2226    2259    2260    2263    2264    2652    2677    2679    2691
         2679    2962    3518    3530    5029    6804   10334   10346   10629   10634   10639   10651   10709   10714   10719
        10732   11130   11144   11152   11159   11619   11625   11634
         4981   11287
        11513
        11206   11222   11289   11382   11462   11571   11940
         1553    1574    1614    1616    1693    1700    1764    1777    1820    1877    1916    1919    1923    1933
         2110    2149    2205    2207    2209    2504    2506    2575    2577    2600    2602    2604    2653    2869
         2880    2888    2890    2892    2983    3150    3207    3303    3350    3383    3500    3502    3504    3514
         3977    4314    4316    4318    4449    4572    4736    4986    9293    8895    9375    9636    9816    9819
        10385   10413   10560   10630   10632   10635   10637   10710   10712   10715   10717   10763   10794   10949
        11028   11059   11131   11160   11211   11268   11276   11284   11298   11305   11352   11359   11364   11369
        11389   11399   11405   11433   11437   11443   11450   11457   11500   11517   11523   11626   11664   11927
        11858   11930   12016
        11227   11262   11302   11373   11401   11427   11631   11929
BR       1515    1518    1555    1575    1580    1584    1589    1593    1597    1602    1606    1618    1661    1694    1723
         1707    1766    1793    1797    1801    1805    1852    1858    1864    1867    1936    1940    1944    1960    1982
         1969    1978    1984    1990    2014    2023    2026    2029    2032    2035    2065    2069    2073    2079    2111
         2153    2210    2216    2222    2248    2254    2605    2637    2643    2663    2669    3683    3693    3813    3911
         2937    2941    2969    2974    3010    3014    3370    3505    3511    3522    3623    3795    3814    3951
```

```
                4319    4345    4349    4540    4606    4612    4987    5054    6808    6815    8294    9821    9901    9907   10169
               10174   10231   10237   10262   10510   10642   10647   10656   10664   10723   10729   10738   10760   10885   10891
               10898   10902   10906   10913   10917   11043   11049   11061   11067   11073   11077   11083   11089   11093   11103
               11135   11139   11142   11155   11158   11208   11225   11264   11281   11291   11300   11307   11367   11446   11473
               11475   11519   11590   11593   11665   11671   11679   11687   11695   11703   11711   11719   11727   11735   11743
               11751   11759   11767   11775   11783   11791   11799   11807   11815   11832   11841   11851   11868   11906   11921
               11942   12008   12032
               9999    9379    9640
               1514    1517    1519    1520    1528    1541    1542    1737    1771    1822    1823    1913    1954    1964    2285
               2731    2948    3142    3295    3346    3572    3969    4441    4616    4728    6683   10242   10243   10281   10282
              10329   10491   10586   10611   10612   10882   11157   11171   11200   11203   11333   11370   11416   11424   11425
              11489   11490   11564   11565   11821   11825   11919   12014
               1781   11156   11229   11280   11306   11524   11589
               1529    1552    1615    1696    1830    1832    1836    1838    1841    1843    1931    1956    1958    1965    1967
               2000    2002    2005    2007    2010    2012    2116    2117    2155    2233    2273    2450    2459    2981    4036
               4045    4228    4237    4342    4513    4529    4796    4805    4916    4925    5198    5207    5345    5354    5507
               5516    5703    5712    5866    5875    6055    6064    6207    6216    6404    6413    6567    6576    6781    6790
               6969    6978    7214    7223    7381    7390    7629    7637    7797    7906    8048    8057    8208    8217    8352
               8361    8604    8613    9215    9224    9462    9471    9727    9736   10260   10353   10519   10527   10659   10668
              10704   10935   10942   10945   11015   11022   11025   11140   11164   11221   11351   11357   11359   11363   11369
              11376   11381   11383   11388   11398   11404   11432   11442   11449   11461   11463   11499   11512   11663   11838
              11940
               1919   11146   11150   11273   11275   11283   11304   11308   11412   11436   11456   11516   11522   11568   11570
               1692    1776    1819    1876    1881    1947    2109    2148    3149    3302    3573    3976    4448    4735    5052
               8292    8894    9374    9635   10256   10285   10384   10412   10559   10631   10636   10711   10716   10762   10793
              10947   10946   11027   11059   11496   11933
              11287   11290   11927   11938
                 61
                174    1943    2225    2257    2299    2404    2595    2646    2672    2909    2944    3048    3154    3264    3540
               3733    3941    4162    4335    4352    4362    4617    4676    5010    5163    5312    5474    5668    5833    6020
               6174    6369    6534    6747    6937    7179    7346    7593    7764    9011    9175    9469    8726    8809    9968
               9063    9182    9300    9429    9556    9693    9864    9910   10177   10396   11109   11263   11632   12007   12031
               1776    1873    1875    2949    3148    3301    3578    3975    4447    4734    8282    8284    8295    8297    8398
               8289    8291   10248   10283   11013   11163   11207   11390   11397   11471   11498   11621   11533   11941   12015
              11980   11168   11310   11615
                 82
               1902     194     195    1576    1714    1784    1788    1809    2213    2507    2510    2531    2580    2609    2695
               2873    2881    2896    3508    3875    3896    4098    4118    4322    4988    5690    6913    9295    9550    9560
               9824   10253   10264   10303   11356   11439   11666
               1572    1761    1909    2061    2096    2135    2195    2243    2292    2296    2306    2339    2349    2391    2397
               2410    2411    2468    2480    2523    2525    2542    2544    2587    2591    2618    2651    2656    2692    2700
               2716    2723    2737    2755    2763    2774    2783    2790    2797    2811    2901    2905    2952    2962    3040
               3044    3056    3090    3111    3160    3171    3215    3227    3256    3260    3340    3343    3353    3364    3374
               3386    3389    3392    3395    3398    3401    3408    3413    3422    3425    3429    3441    3492    3533    3537
               3597    3645    3648    3652    3663    3675    3726    3730    3769    3817    3820    3824    3836    3847    3899
               3991    3908    3910    3933    3937    3993    4056    4059    4066    4069    4077    4111    4112    4130    4132
               4154    4158    4193    4247    4255    4262    4272    4327    4331    4355    4359    4394    4467    4564    4574
               4589    4618    4669    4673    4683    4687    4750    4817    4824    4838    4859    4870    4931    4944    4954
               4960    4976    5002    5006    5011    5034    5087    5095    5105    5117    5124    5133    5152    5159    5219
               5221    5224    5232    5253    5269    5273    5282    5301    5308    5365    5368    5371    5379    5399    5429
               5436    5445    5463    5470    5527    5530    5533    5541    5558    5603    5610    5615    5625    5630    5639
               5657    5664    5723    5726    5729    5737    5756    5773    5777    5782    5790    5795    5805    5822    5828
               5886    5889    5892    5900    5918    5965    5972    5982    5991    6009    6016    6075    6076    6091    6099
               6109    6124    6128    6136    6145    6163    6170    6227    6230    6233    6241    6259    6304    6311    6316
```

```
                6326    6331    6340    6358    6365    6424    6427    6430    6438    6457    6474    6478    6483    6491    6495
                6505    6523    6530    6587    6590    6593    6601    6619    6671    6687    6692    6702    6707    6716    6736
                6743    6824    6827    6830    6838    6858    6875    6879    6884    6892    6897    6906    6926    6933    6939
                6992    6995    7003    7022    7076    7083    7092    7100    7109    7121    7128    7134    7141    7150    7168
                7176    7234    7237    7240    7248    7269    7285    7289    7298    7307    7316    7335    7342    7401    7404
                7407    7415    7436    7489    7496    7505    7513    7522    7534    7541    7547    7554    7563    7582    7589
                7648    7651    7654    7657    7660    7668    7689    7705    7709    7717    7726    7734    7753    7760    7816
                7819    7822    7829    7850    7888    7935    7943    7953    7962    7972    7981    8000    8007    8068    8071
                8076    8076    8081    8089    8108    8112    8122    8130    8135    8144    8164    8171    8228    8231    8234
                8242    8260    8304    8309    8313    8372    8380    8387    8396    8419    8427    8434    8439    8447    8464
                8495    8528    8561    8565    8624    8632    8639    8648    8672    8676    8685    8692    8697    8705    8722
                8753    8801    8805    8813    8841    8848    8859    8864    8869    8874    8879    8909    8915    8922    8930
                8939    8957    8964    8994    8997    9000    9008    9028    9033    9052    9059    9090    9093    9096    9105
                9126    9146    9151    9171    9178    9235    9238    9241    9249    9292    9296    9305    9332    9339    9344
                9349    9354    9359    9368    9393    9401    9418    9425    9482    9485    9488    9496    9541    9552    9566
                9593    9600    9605    9610    9615    9620    9648    9653    9661    9682    9689    9747    9750    9753    9760
                9829    9836    9844    9860    9893    9914    9924    9948    9952    9960    9977    10016   10024   10028   10035
                10052   10075   10081   10088   10096   10137   10159   10181   10192   10298   10333   10345   10351   10830   10840
                11021   11052   11282   11289   11296   11354   11415   11460   11627
MCY             1513    1516    1527    1531    1533    1534    1535    1536    1537    1538    1539    1540    1544    1545    1548
                1549    1550    1551    1556    1558    1559    1560    1565    1566    1567    1568    1569    1610    1624    1652
                1653    1666    1667    1672    1682    1683    1684    1685    1657    1689    1690    1691    1695    1697    1711
                1716    1718    1728    1729    1730    1732    1733    1736    1738    1758    1759    1769    1770    1772    1773
                1774    1816    1817    1819    1821    1824    1925    1829    1855    1861    1869    1871    1979    1980    1896
                1904    1905    1906    1907    1917    1928    1946    1949    1955    1961    1970    1981    1987    1993    2045
                2046    2060    2093    2094    2101    2102    2103    2105    2106    2108    2113    2121    2132    2133    2140
                2141    2142    2144    2145    2147    2152    2158    2192    2193    2199    2219    2228    2231    2236    2251
                2266    2270    2276    2284    2286    2300    2313    2321    2324    2372    2394    2395    2405    2419    2426
                2429    2433    2434    2447    2449    2453    2454    2457    2458    2462    2463    2520    2521    2524    2539
                2540    2543    2571    2584    2585    2640    2650    2666    2834    2837    2841    2842    2864    2898    2899
                2923    2934    2960    2971    2992    2995    2999    3000    3037    3038    3050    3063    3071    3074    3115
                3116    3118    3121    3123    3125    3127    3129    3131    3133    3137    3138    3139    3141    3143    3146
                3151    3152    3153    3157    3165    3178    3186    3189    3253    3254    3268    3269    3271    3274    3276
                3278    3280    3282    3284    3286    3290    3291    3292    3294    3296    3299    3304    3305    3306    3310
                3313    3314    3322    3325    3404    3407    3417    3419    3489    3490    3515    3526    3541    3546    3548
                3551    3553    3555    3557    3559    3561    3563    3567    3568    3569    3571    3573    3576    3581    3582
                3583    3587    3590    3591    3604    3614    3617    362.    3622    362?    3630    3698    3723    3724    3734
                3739    3741    3744    3745    3748    3750    3752    3754    3756    3758    3759    3762    3763    3776    3786
                3789    3793    3794    3799    3802    3886    3887    3890    3905    3906    3909    3930    3931    3942    3943
                3945    3948    3950    3952    3954    3956    3958    3960    3964    3965    3966    3968    3970    3973    3978
                3979    3980    3984    3987    3999    4006    4009    4012    4013    4025    4029    4035    4039    4040    4043
                4044    4048    4049    4108    4109    4112    4127    4128    4131    4151    4152    4163    4165    4167    4170
                4172    4174    4176    4178    4180    4182    4184    4185    4187    4199    4206    4209    4212    4213    4220
                4227    4231    4232    4235    4236    4240    4241    4324    4325    4340    4367    4370    4390    4395    4399
                4405    4410    4414    4418    4422    4426    4430    4436    4437    4438    4440    4442    4445    4450    4451
                4458    4458    4461    4474    4482    4485    4489    4490    4502    4506    4512    4516    4517    4522    4530
                4532    4532    4533    4553    4554    4560    4609    4628    4631    4635    4636    4665    4666    4667    4680
                4694    4697    4702    4705    4708    4711    4715    4717    4719    4723    4724    4725    4727    4729    4730
                4737    4738    4739    4743    4745    4757    4765    4768    4771    4772    4795    4799    4795    4799    4900
                4803    4804    4808    4809    4851    4864    4877    4885    4888    4891    4892    4905    4909    4915    4919
                4920    4923    4924    4928    4929    4997    4999    5000    5016    5019    5039    5042    5084    5095    5165
                5173    5176    5179    5180    5195    5197    5201    5202    5205    5206    5210    5211    5314    5322    5328
                5328    5329    5342    5344    5348    5349    5352    5353    5357    5358    5426    5427    5476    5494    5498
                5490    5491    5504    5506    5510    5511    5514    5515    5519    5520    5600    5601    5670    5679    5691
```

```
 5584  5695  5700  5702  5706  5707  5710  5711  5715  5716  5835  5843  5846  5849  5850
 5863  5865  5869  5870  5873  5874  5878  5879  5962  5963  6022  6030  6033  6036  6037
 6052  6054  6058  6059  6062  6063  6067  6068  6176  6184  6187  6190  6191  6204  6206
 6210  6211  6214  6215  6219  6220  6301  6302  6371  6379  6382  6385  6396  6401  6403
 6407  6409  6411  6412  6416  6417  6536  6544  6547  6550  6551  6564  6566  6570  6571
 6574  6575  6579  6580  6667  6668  6669  6749  6756  6759  6762  6763  6778  6780  6784
 6795  6798  6799  6793  6794  6806  6939  6946  6949  6952  6953  6966  6968  6972  6973
 6976  6977  6981  6982  7073  7074  7181  7189  7192  7195  7196  7211  7213  7217  7218
 7224  7226  7226  7225  7348  7356  7359  7362  7363  7378  7380  7384  7385  7388  7389
 7393  7394  7486  7487  7595  7603  7606  7609  7610  7625  7627  7631  7632  7635  7636
 7640  7641  7766  7774  7777  7780  7781  7794  7796  7800  7801  7804  7805  7809  7910
 7885  7886  7898  7899  7900  7901  7902  7903  7904  7905  7906  7907  7911  7918  7921
 7924  7925  7933  8013  8021  8024  8027  8028  8045  8047  8051  8052  8055  8056  8060
 8061  8177  8185  8188  8191  8192  8205  8207  8211  8212  8215  8216  8220  8221  8320
 8328  8331  8334  8335  8349  8351  8355  8356  8359  8360  8364  9365  9470  9478  8481
 8525  8526  8534  8542  8545  8548  6549  8572  8580  8583  8586  3587  8601  8603  8607
 8608  8611  8612  8616  8617  8729  8736  8739  8798  8799  8816  8824  8927  8830  9831
 8887  8888  8889  8890  8892  8900  8901  8904  8970  8978  8991  9065  9073  9076  9184
 9192  9195  9198  9199  9212  9214  9218  9219  9222  9223  9227  9229  9289  9290  9308
 9316  9319  9367  9368  9369  9370  9372  9380  9381  9384  9431  9439  9442  9445  9446
 9459  9461  9465  9466  9469  9470  9474  9475  9538  9539  9569  9577  9580  9629  9629
 9630  9631  9633  9641  9642  9645  9695  9704  9707  9710  9711  9724  9726  9730  9731
 9734  9735  9739  9740  9826  9827  9866  9874  9877  9904  9945  9984  9992  9995 10009
10010  10011 10112 10120 10123 10171 10210 10227 10228 10234 10240 10250 10255 10258 10259
10263  10288 10292 10295 10324 10325 10327 10328 10330 10331 10352 10355 10356 10360 10376
10377  10378 10379 10383 10386 10387 10405 10406 10407 10408 10409 10410 10411 10414 10415
10416  10433 10434 10435 10436 10438 10439 10440 10452 10453 10454 10470 10471 10480 10484
10485  10486 10487 10489 10490 10506 10508 10512 10514 10516 10521 10529 10552 10553 10554
10555  10556 10557 10558 10561 10562 10563 10587 10588 10589 10613 10614 10615 10616 10617
10618  10619 10621 10622 10623 10624 10625 10628 10633 10638 10650 10674 10675 10676 10677
10697  10698 10699 10700 10702 10703 10705 10709 10713 10718 10734 10740 10741 10752 10753
10754  10755 10764 10765 10782 10783 10784 10785 10786 10787 10788 10789 10791 10795 10796
10797  10798 10819 10820 10821 10822 10823 10824 10825 10826 10827 10828 10832 10837 10838
10839  10843 10851 10852 10853 10854 10855 10893 10888 10894 10911 10921 10922 10939 10940
10941  10942 10943 10944 10945 10948 10949 10950 10968 10969 10970 10971 10972 10974 10975
10977  11002 11003 11004 11005 11006 11007 11008 11009 11010 11011 11012 11014 11018 11019
11029  11030 11031 11032 11033 11034 11046 11054 11055 11064 11071 11090 11086 11107 11135
11136  11138 11141 11154 11166 11167 11169 11170 11173 11174 11190 11191 11192 11193 11194
11195  11196 11201 11204 11224 11230 11231 11232 11233 11234 11236 11237 11265 11266 11270
11285  11334 11335 11336 11337 11339 11378 11390 11421 11438 11453 11458 11487 11488 11491
11509  11510 11511 11526 11527 11528 11529 11556 11557 11558 11559 11560 11562 11563 11582
11583  11584 11585 11586 11613 11617 11622 11636 11639 11661 11674 11682 11690 11698 11706
11721  11722 11730 11738 11746 11754 11762 11770 11778 11786 11794 11802 11810 11818 11824
11829  11842 11843 11848 11853 11854 11856 11959 11863 11870 11871 11902 11910 11911 11912
11918  11925 11943 11944 11945 11946 11947 11962 11963 11967 11996 11997 11998 11999 12000
12001  12002 12003 12004 12005 12006 12012 12013 12017 12018 12019 12020 12021 12022 12023
12024  12025 12028
MC.B    1543   1619   1779   1976   4677   4678  10831  11172  11199  11202  11216  11219  11228  11267  11295
       11303  11349  11374  11386  11402  11417  11429  11497  11515  11520  11566  11624  11903  11904  11907
       11908  11909  11913  11916  11917  11936  11965
MSG    11980  11914
MSP     4385   4386   4387   4619   4620   4621   4688   4689   4690   4860   4861   4862  10299  10300  10301
       10340  10341  10342  10343  10731
RESET   1524   1760   2390  10297
```

CROSS REFERENCE TABLE

```
RCL      9893  9373  9634 11573 11575 11577 11920 11922 11923 11924 11926
RCR      1826  3144  3297  3574  3971  4443  4730
RTI      1557 10590 10678 10742 10923 11175 11238 11272 11379 11391 11459 11492 11502 11530 11587
        11641 11948 12030
RTS      2358  2489  2747  2820  3099  3236  3450  3684  3856  4086  4281  4598  4847  4968  5241
         5262  5388  5408  5550  5567  5746  5765  5909  5927  6098  6117  6250  6268  6447  6466
         6610  6628  6848  6968  7013  7032  7257  7278  7425  7446  7677  7638  7839  7860  8097
         9251  8269  9405  8504  8657  9762  9017  9115  9136  9258  9504  9770  9933 10061 10201
        10361 10388 10417 10441 10456 10472 10481 10492 10533 10538 10564 10766 10799 10856 10951
        10978 11035 11312 11340 11872 11968
SEC      9891  9371  9632
SOB     10326 10437 10507 10513 10620 10657 10701 10756 11020 11205 11623
SOB     11970 11979 11980 11981 11982 11984 11986 11987 11988 11989 11990
TST      1573  1613  1658  1884  1915  1921  1929  2043  2044  2204  2206  2208  2282  2503  2505
         2574  2576  2599  2601  2603  2867  2869  2887  2869  2891  3154  3204  3206  3307  3349
         3382  3499  3501  3503  3584  3611  3793  3872  3981  4027  4095  4313  4315  4317  4453
         4504  4571  4740  4787  4907  6678  9548  9558  9815  9817  9819 10244 10757 11137 11161
        11210 11220 11269 11277 11299 11338 11355 11366 11371 11414 11451 11466 11494 11581 11588
        11630 11637 11865 11931 11964
TSTB     1734  1782  4984 11148 11212 11226 11261 11301 11372 11400 11426 11957
.ASCII    299   300 12161 12179 12195 12213 12247 12265 12283 12313 12323 12349 12359 12387 12395
        12415 12424 12448 12456 12474 12482 12498 12510 12529 12542 12560 12569 12581 12584 12600
        12612 12619 12633 12644 12650 12666 12679 12689 12711 12721 12737 12747 12766 12779 12805
        12819 12835 12845 12851 12867 12878 12891 12900 12905 12925 12937 12947 12998 13007 13026
        13036 13047 13056 13063 13080 13086 13090 13105 13111 13117 13134 13138 13146 13162 13170
        13174 13193 13199 13211 13215 13223 13239 13311 13318 13327 13344 13364 13381 13395 13413
        13433 13455 13470 13487 13504 13515 13526 13552
.ASCIZ    298   301  1582  1586  1591  1595  1599  1604  1608  1663  1705  1709  1768  1795  1799
         1803  1807  1854  1860  1866  1938  1942  1980  1986  1992  2025  2031  2037  2067  2071
         2075  2091  2218  2224  2250  2256  2639  2645  2665  2671  2915  2919  2933  2939  2943
         2970  2976  3012  3016  3513  3524  4347  4351  4542  4609  4614  6810  6818  9903  9909
        10170 10176 10233 10239 10306 10887 10893 10900 10904 10908 10915 10919 11045 11051 11063
        11069 11075 11079 11085 11091 11095 11105 11534 11535 11536 11537 11539 11673 11681 11689
        11697 11705 11713 11721 11729 11737 11745 11753 11761 11769 11777 11785 11793 11801 11809
        11817 11873 12034 12046 12050 12060 12069 12082 12089 12098 12108 12115 12127 12139 12153
        12169 12185 12202 12219 12231 12239 12253 12271 12291 12303 12330 12340 12366 12375 12402
        12411 12431 12440 12463 12489 12517 12549 12593 12630 12657 12677 12699 12728 12754 12785
        12796 12826 12858 12889 12912 12959 12972 12981 12990 13014 13023 13070 13095 13124 13153
        13166 13190 13203 13230 13252 13265 13274 13283 13291 13300 13336 13354 13373 13389 13401
        13406 13423 13444 13462 13478 13496 13510 13521 13533 13540 13560
.BLKB    11321 11533
.BLKW     1435  1436  1453  1472 11243
.BYTE      254   255   260   261   269   270   278   279   280   281  1464  5135  5136  5294  5285
         5447  5448  5641  5642  5806  5807  5993  5994  6147  6148  6342  6343  6507  6509  6719
         6719  6908  6909  7152  7153  7318  7319  7565  7566  7736  7737  7983  7984  9146  9147
         8449  8450  8707  8708  8940  8941  9035  9036  9153  9154  9403  9404  9663  9664  9846
         9847  9962  9963 10098 10099 10305 11531 11532 11949 11950 11951 11952 13614 13617 13620
        13622 13624 13626 13630 13634 13637 13640 13642 13644 13646 13648 13652
.DSABL   11475
.ENABL       2 11317
.END     13658
.ENDC       7    17    25    33    39    48    50    51    52    61   153   167   177   181   185
          187   220   231   242   244   248   252   254   282   290   296   297   298   299   303
         1090  1092  1132  1134  1375  1380  1434  1524  1531  1532  1535  1537  1539  1541  1543
```

1544	1546	1562	1582	1586	1591	1595	1599	1604	1609	1615	1621	1627	1649	1663
1675	1677	1680	1681	1682	1693	1705	1709	1722	1723	1726	1727	1728	1729	1745
1746	1756	1757	1758	1759	1768	1795	1799	1803	1807	1836	1846	1854	1860	1866
1891	1892	1902	1903	1904	1905	1906	1938	1942	1954	1973	1980	1986	1992	2000
2025	2031	2036	2039	2052	2053	2058	2059	2060	2067	2071	2075	2081	2084	2085
2090	2091	2092	2125	2126	2129	2130	2131	2165	2166	2190	2191	2192	2218	2224
2250	2256	2324	2328	2336	2338	2344	2378	2379	2385	2386	2387	2429	2433	2435
2449	2467	2468	2474	2503	2513	2517	2518	2519	2520	2536	2537	2538	2539	2553
2554	2569	2570	2571	2572	2574	2583	2639	2645	2665	2671	2692	2700	2702	2704
2705	2713	2715	2716	2722	2723	2729	2763	2771	2773	2774	2782	2783	2789	2790
2796	2797	2803	2837	2841	2843	2850	2855	2956	2862	2863	2864	2865	2967	2976
2915	2919	2933	2939	2943	2970	2976	2995	2999	3001	3008	3012	3016	3029	3030
3035	3036	3037	3074	3078	3086	3116	3118	3120	3121	3123	3125	3127	3129	3131
3133	3135	3157	3159	3189	3193	3201	3214	3215	3221	3246	3247	3251	3252	3253
3269	3271	3273	3274	3276	3278	3280	3282	3284	3286	3289	3310	3312	3325	3329
3337	3353	3361	3363	3364	3370	3374	3382	3429	3435	3461	3462	3487	3488	3489
3513	3524	3546	3548	3550	3551	3553	3555	3557	3559	3561	3563	3565	3587	3589
3617	3621	3623	3628	3630	3632	3633	3642	3652	3660	3662	3663	3669	3702	3703
3721	3722	3723	3739	3741	3743	3744	3746	3748	3750	3752	3754	3756	3758	3759
3761	3789	3793	3795	3800	3802	3804	3805	3814	3824	3832	3834	3835	3841	3872
3878	3883	3884	3885	3886	3902	3903	3904	3905	3921	3922	3928	3929	3930	3943
3945	3947	3948	3950	3952	3954	3956	3958	3960	3962	3984	3986	4007	4009	4011
4012	4014	4035	4053	4066	4095	4101	4105	4106	4107	4108	4124	4125	4126	4127
4142	4143	4149	4150	4151	4165	4167	4169	4170	4172	4174	4176	4178	4180	4192
4184	4185	4187	4207	4209	4211	4212	4214	4227	4245	4254	4255	4261	4262	4268
4288	4289	4307	4308	4309	4347	4351	4370	4374	4382	4395	4397	4399	4401	4404
4405	4408	4410	4412	4414	4416	4418	4420	4422	4424	4426	4428	4430	4432	4456
4458	4460	4485	4489	4491	4542	4564	4570	4574	4580	4584	4608	4614	4631	4635
4637	4644	4646	4647	4663	4664	4665	4666	4694	4696	4697	4699	4701	4702	4704
4705	4707	4708	4710	4711	4713	4717	4719	4721	4743	4745	4766	4768	4770	4771
4773	4795	4813	4817	4823	4824	4830	4834	4858	4886	4888	4890	4891	4893	4915
4933	4937	4943	4944	4950	4954	4991	4992	4995	4996	4997	5013	5014	5017	5019
5021	5022	5029	5040	5042	5044	5045	5052	5058	5059	5082	5083	5084	5099	5100
5101	5102	5121	5122	5123	5124	5139	5143	5144	5145	5146	5174	5176	5178	5179
5197	5197	5215	5288	5292	5293	5294	5295	5323	5325	5327	5328	5330	5344	5362
5415	5416	5424	5425	5426	5450	5454	5455	5456	5457	5485	5487	5489	5490	5492
5526	5524	5572	5573	5598	5599	5600	5644	5648	5649	5650	5651	5679	5681	5693
5694	5686	5702	5720	5809	5813	5814	5815	5816	5844	5846	5848	5849	5851	5865
5883	5934	5935	5960	5961	5962	5996	6000	6001	6002	6003	6031	6033	6035	6036
6038	6054	6072	6150	6154	6155	6156	6157	6195	6197	6199	6190	6192	6206	6224
6273	6274	6293	6300	6301	6345	6349	6350	6351	6352	6380	6382	6384	6385	6397
6403	5421	6510	6514	6515	6516	6517	6545	6547	6549	6550	6552	6566	6594	6636
6637	6665	6666	6667	6668	6721	6725	6726	6727	6728	6757	6759	6761	6762	6764
6780	6798	6810	6818	6911	6915	5916	6917	6918	6947	6949	6951	6952	6954	6969
5985	7043	7044	7071	7072	7073	7087	7088	7089	7090	7104	7105	7106	7107	7125
7126	7127	7128	7138	7139	7140	7141	7156	7160	7161	7162	7163	7190	7192	7194
7195	7197	7213	7231	7302	7303	7304	7305	7306	7307	7323	7326	7327	7328	7329
7357	7359	7361	7362	7364	7380	7398	7452	7453	7484	7485	7486	7500	7501	7502
7503	7517	7518	7519	7520	7538	7539	7540	7541	7551	7552	7553	7554	7569	7573
7574	7575	7576	7604	7606	7608	7609	7611	7627	7645	7721	7722	7723	7724	7725
7726	7740	7744	7745	7746	7747	7775	7777	7779	7780	7782	7796	7814	7867	7968
7983	7884	7985	7919	7921	7923	7924	7926	7933	7947	7948	7949	7950	7966	7967
7988	7969	7987	7991	7992	7993	7994	9022	8024	8026	8027	8029	8047	9065	9126
9127	9128	9129	8150	8154	8155	8156	8157	8196	8198	8190	8191	8193	9207	9235

DZRJIA.P11 CROSS REFERENCE TABLE

```
          9329  9331  9333  9334  9336  8351  8369  8379  8380  2386  8387  8393  8431  8432  8433
          9434  8453  8457  8459  9459  8460  8479  8481  8483  8484  8492  8513  8514  8523  8524
          9525  8543  9545  8547  8548  8550  8558  8581  8583  8585  8586  8588  8603  8621  8631
          8632  8638  9539  8645  8689  8690  8691  8692  9711  8715  8716  8717  8718  8737  8739
          8741  8742  8750  8773  8774  8796  8797  8798  8825  8827  8829  8830  8832  8840  8852
          8953  8954  8955  8919  8920  8921  8922  8944  8948  8949  8950  8551  8979  8981  8983
          8984  8992  9039  9043  9044  9045  9046  9074  9076  9078  9079  9087  9157  9161  9162
          9163  9164  9193  9195  9197  9198  9200  9214  9232  9269  9270  9287  9288  9289  9317
          9319  9321  9322  9330  9406  9410  9411  9412  9413  9440  9442  9444  9445  9447  9461
          9479  9516  9517  9536  9537  9538  9578  9580  9582  9593  9591  9666  9670  9671  9672
          9673  9705  9707  9709  9710  9712  9726  9744  9785  9786  9809  9810  9811  9849  9853
          9854  9855  9856  9875  9877  9879  9880  9889  9903  9909  9914  9920  9965  9969  9970
          9971  9972  9993  9995  9997  9998 10006 10016 10024 10028 10034 10035 10043 10047 10101
         10105 10106 10107 10108 10121 10123 10125 10126 10134 10170 10176 10181 10187 10214 10215
         10216 10217 10225 10226 10227 10228 10233 10239 10274 10275 10276 10278 10281 10287 10290
         10291 10295 10297 10303 10305 10306 10309 10887 10893 10900 10904 10908 10915 10919 11045
         11051 11063 11069 11075 11079 11085 11091 11095 11100 11105 11110 11117 11120 11125 11130
         11132 11143 11146 11147 11148 11150 11152 11159 11163 11168 11169 11173 11176 11177 11180
         11247 11267 11317 11348 11357 11361 11392 11394 11409 11440 11476 11480 11491 11503 11504
         11511 11513 11516 11518 11534 11535 11541 11545 11551 11595 11598 11601 11615 11622 11627
         11628 11629 11630 11641 11642 11673 11681 11689 11697 11705 11713 11721 11729 11737 11745
         11753 11761 11769 11777 11785 11793 11801 11809 11817 11880 11957 11963 11966 11978 11979
         11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11995 12004 12005 12011
         12017 12018 12028 12030 12037
.EQUIV     61    62    70    85    86   115   116   117   118   119   120   121   122   123   124
          143   144   145   146   147   148   149   150   151   152
.EVEN     1582  1586  1591  1595  1599  1604  1608  1663  1705  1709  1768  1795  1799  1803  1807
          1954  1960  1866  1938  1942  1980  1986  1992  2025  2031  2037  2067  2071  2075  2081
          2218  2224  2250  2256  2639  2645  2665  2671  2915  2919  2933  2939  2943  2970  2976
          3012  3016  3513  3524  4347  4351  4542  4608  4614  6810  6818  9903  9909 10170 10176
         10233 10239 10887 10893 10900 10904 10908 10915 10919 11045 11051 11063 11069 11075 11079
         11085 11091 11095 11105 11323 11673 11681 11689 11697 11705 11713 11721 11729 11737 11745
         11753 11761 11769 11777 11785 11793 11801 11809 11817 11874 12036 13569 13655
.IF           3    14    22    32    38    47    49    50    51    52    59   125   153   177   190
          193   185   220   231   242   243   247   251   253   282   290   296   297   298   302
          333  1089  1091  1131  1133  1374  1379  1434  1524  1526  1531  1533  1535  1537  1539
         1541  1542  1544  1562  1581  1585  1590  1594  1598  1603  1607  1612  1615  1626  1648
         1652  1675  1677  1680  1682  1683  1704  1708  1721  1723  1726  1728  1729  1744  1746
         1756  1758  1759  1767  1794  1798  1802  1806  1835  1845  1853  1859  1865  1890  1892
         1902  1904  1905  1906  1937  1941  1953  1972  1979  1985  1991  1999  2024  2030  2036
         2038  2051  2053  2058  2060  2066  2070  2074  2080  2083  2085  2090  2092  2124  2126
         2129  2131  2164  2166  2190  2192  2217  2223  2249  2255  2322  2324  2326  2328  2336
         2338  2344  2377  2379  2385  2387  2427  2429  2431  2433  2443  2449  2468  2474  2502
         2512  2516  2518  2520  2535  2537  2539  2552  2554  2569  2571  2572  2573  2592  2638
         2644  2664  2670  2692  2698  2700  2702  2704  2705  2711  2713  2715  2716  2722  2723
         2729  2763  2769  2771  2773  2774  2780  2782  2783  2789  2790  2796  2797  2803  2835
         2837  2839  2841  2850  2854  2856  2862  2864  2865  2866  2875  2914  2919  2932  2939
         2942  2969  2975  2993  2995  2997  2999  3008  3011  3015  3028  3030  3035  3037  3072
         3074  3076  3078  3086  3116  3118  3120  3121  3123  3125  3127  3129  3131  3133  3135
         3157  3187  3189  3191  3193  3201  3212  3214  3215  3221  3245  3247  3251  3253  3269
         3271  3273  3274  3276  3278  3280  3282  3284  3286  3288  3310  3323  3325  3327  3329
         3337  3353  3359  3361  3363  3364  3370  3374  3380  3382  3429  3435  3460  3462  3487
         3489  3512  3523  3546  3548  3550  3551  3553  3555  3557  3559  3561  3563  3565  3597
         3615  3617  3619  3621  3628  3632  3633  3642  3652  3658  3660  3662  3663  3669  3701
```

CROSS REFERENCE TABLE

3703	3721	3723	3739	3741	3743	3744	3746	3748	3750	3752	3754	3756	3758	3759
3787	3789	3791	3793	3800	3804	3805	3814	3824	3830	3832	3834	3835	3841	3871
3877	3882	3884	3886	3901	3903	3905	3920	3922	3928	3930	3943	3945	3947	3948
3950	3952	3954	3956	3958	3960	3962	3984	4007	4011	4012	4021	4027	4063	4066
4094	4100	4104	4106	4108	4123	4125	4127	4141	4143	4149	4151	4165	4167	4169
4170	4172	4174	4176	4178	4180	4182	4184	4185	4207	4211	4212	4221	4227	4251
4254	4255	4261	4262	4268	4287	4289	4307	4309	4346	4350	4368	4370	4372	4374
4382	4395	4399	4403	4404	4406	4410	4414	4418	4422	4426	4430	4434	4458	4483
4485	4487	4489	4541	4564	4570	4574	4580	4582	4584	4607	4613	4629	4631	4633
4635	4644	4645	4647	4663	4665	4666	4694	4697	4700	4701	4702	4705	4708	4711
4717	4719	4721	4743	4766	4770	4771	4781	4787	4817	4823	4824	4830	4831	4834
4857	4886	4890	4891	4901	4907	4937	4943	4944	4950	4951	4954	4990	4992	4995
4997	5013	5017	5021	5022	5029	5040	5044	5045	5052	5057	5059	5082	5084	5098
5099	5100	5101	5102	5120	5121	5122	5123	5124	5137	5139	5142	5143	5144	5145
5174	5178	5179	5191	5197	5286	5291	5292	5293	5294	5323	5327	5328	5338	5344
5414	5416	5424	5426	5449	5453	5454	5455	5456	5485	5489	5490	5500	5506	5571
5573	5598	5600	5643	5647	5648	5649	5650	5679	5683	5684	5696	5702	5808	5812
5813	5814	5815	5844	5848	5849	5859	5865	5933	5935	5960	5962	5995	5999	6000
6001	6002	6031	6035	6036	6048	6054	6149	6153	6154	6155	6155	6185	6189	6190
6200	6206	6272	6274	6299	6301	6344	6348	6349	6350	6351	6380	6384	6395	6397
6403	6509	6513	6514	6515	6516	6545	6549	6550	6560	6566	6635	6637	6665	6667
6668	6720	6724	6725	6726	6727	6757	6761	6762	6774	6780	6809	6817	6910	6914
6915	6916	6917	6947	6951	6952	6962	6968	7042	7044	7071	7073	7086	7087	7088
7089	7090	7103	7104	7105	7106	7107	7124	7125	7126	7127	7128	7137	7138	7139
7140	7141	7154	7156	7159	7160	7161	7162	7190	7194	7195	7207	7213	7301	7302
7303	7304	7305	7306	7307	7320	7325	7326	7327	7328	7357	7361	7362	7374	7380
7451	7453	7484	7486	7499	7500	7501	7502	7503	7516	7517	7518	7519	7520	7537
7538	7539	7540	7541	7550	7551	7552	7553	7554	7567	7569	7572	7573	7574	7575
7604	7608	7609	7621	7627	7720	7721	7722	7723	7724	7725	7726	7738	7743	7744
7745	7746	7775	7779	7780	7790	7796	7866	7868	7883	7885	7919	7923	7924	7933
7946	7947	7948	7949	7950	7965	7966	7967	7968	7969	7985	7987	7990	7991	7992
7993	8022	8026	8027	8041	8047	8125	8126	8127	8128	8129	8148	8153	8154	8155
8156	8186	8190	8191	8201	8207	8329	8333	8334	8345	8351	8376	8379	8380	8396
8387	8393	8430	8431	8432	8433	8434	8451	8456	8457	8458	8459	8479	8483	8494
8492	8512	8514	8523	8525	8543	8547	8548	8558	8581	8585	8586	8597	8603	8628
8631	8632	8638	8639	8645	8688	8689	8690	8691	8692	8709	8714	8715	8716	8717
8737	8741	8742	8750	8772	8774	8796	8798	8825	8829	8830	8940	6851	8852	8953
8854	8855	8918	8919	8920	8921	8922	8942	8944	8947	8948	8949	8950	8979	8983
8984	8992	9037	9042	9043	9044	9045	9074	9078	9079	9087	9155	9157	9160	9161
9162	9163	9193	9197	9198	9208	9214	9268	9270	9287	9289	9317	9321	9322	9330
9405	9409	9410	9411	9412	9440	9444	9445	9455	9461	9515	9517	9536	9538	9579
9582	9583	9591	9665	9669	9670	9671	9672	9705	9709	9710	9720	9726	9784	9796
9809	9811	9848	9852	9853	9854	9855	9875	9879	9880	9889	9902	9908	9914	9920
9964	9968	9969	9970	9971	9993	9997	9998	10006	10016	10022	10024	10028	10034	10035
10041	10043	10044	10047	10100	10104	10105	10106	10107	10121	10125	10126	10134	10169	10175
10181	10187	10213	10214	10215	10217	10225	10227	10228	10232	10238	10273	10274	10275	10276
10277	10278	10280	10286	10289	10291	10295	10297	10303	10305	10306	10886	10892	10999	10903
10907	10914	10918	11044	11050	11062	11068	11074	11078	11084	11090	11094	11099	11104	11109
11116	11119	11124	11130	11142	11144	11145	11146	11148	11149	11150	11159	11161	11169	11170
11175	11176	11177	11179	11246	11267	11316	11318	11346	11351	11357	11393	11394	11408	11432
11479	11480	11490	11503	11511	11512	11516	11517	11533	11534	11541	11544	11547	11563	11597
11600	11611	11618	11625	11627	11628	11630	11634	11641	11642	11672	11680	11688	11696	11704
11712	11720	11728	11736	11744	11752	11760	11768	11776	11784	11792	11800	11808	11816	11979
11956	11962	11966	11970	11979	11980	11981	11982	11983	11984	11986	11987	11998	11989	11990

DZRJIA.P11 CROSS REFERENCE TABLE

	11994	12004	12005	12010	12017	12018	12026	12028	12030	12034					
.IFF	33	39	47	50	51	52	61	181	185	244	248	251	253	282	303
	1090	1092	1132	1134	1375	1380	1531	1627	1649	1676	1677	1681	1682	1683	1722
	1723	1727	1728	1729	1745	1746	1757	1758	1759	1836	1846	1891	1892	1903	1904
	1905	1954	1973	2000	2039	2052	2053	2059	2060	2084	2085	2091	2092	2125	2126
	2130	2131	2165	2166	2191	2192	2338	2378	2379	2386	2387	2468	2503	2513	2517
	2518	2519	2520	2536	2537	2538	2539	2553	2554	2570	2571	2572	2574	2583	2692
	2705	2716	2723	2763	2774	2783	2790	2797	2855	2856	2863	2864	2865	2867	2876
	3029	3030	3036	3037	3116	3118	3121	3123	3125	3127	3129	3131	3133	3135	3157
	3212	3214	3215	3246	3247	3252	3253	3269	3271	3274	3276	3278	3280	3282	3284
	3286	3288	3310	3353	3364	3374	3429	3461	3462	3488	3489	3546	3548	3551	3553
	3555	3557	3559	3561	3563	3565	3587	3652	3663	3702	3703	3722	3723	3739	3741
	3744	3746	3748	3750	3752	3754	3756	3758	3759	3824	3835	3872	3878	3883	3884
	3885	3886	3902	3903	3904	3905	3921	3922	3929	3930	3943	3945	3948	3950	3952
	3954	3956	3958	3960	3962	3984	4066	4095	4101	4105	4106	4107	4108	4124	4125
	4126	4127	4142	4143	4150	4151	4165	4167	4170	4172	4174	4176	4178	4180	4192
	4184	4185	4254	4255	4262	4288	4289	4308	4309	4395	4399	4404	4406	4410	4414
	4418	4422	4426	4430	4434	4456	4459	4564	4574	4582	4584	4646	4647	4664	4665
	4694	4697	4701	4702	4705	4708	4711	4717	4719	4721	4743	4817	4824	4834	4858
	4937	4944	4954	4991	4992	4996	4997	5058	5059	5083	5084	5137	5139	5143	5144
	5145	5286	5288	5292	5293	5294	5415	5416	5425	5426	5449	5454	5455	5456	5572
	5573	5599	5600	5643	5648	5649	5650	5808	5813	5814	5815	5934	5935	5961	5962
	5995	6000	6001	6002	6149	6154	6155	6156	6273	6274	6300	6301	6344	6349	6350
	6351	6509	6514	6515	6516	6636	6637	6666	6667	6668	6720	6725	6726	6727	6910
	6915	6916	6917	7043	7044	7072	7073	7154	7156	7160	7161	7162	7320	7322	7326
	7327	7328	7452	7453	7485	7486	7567	7569	7573	7574	7575	7738	7740	7744	7745
	7746	7867	7868	7884	7885	7985	7987	7991	7992	7993	8148	8150	8154	8155	8156
	9379	8380	8387	8451	8453	8457	8458	8459	8513	8514	8524	8525	8631	8632	8639
	9709	8711	8715	8716	8717	8773	8774	8797	8738	8942	8944	8948	8949	8950	9037
	9039	9043	9044	9045	9157	9161	9162	9163	9269	9270	9288	9289	9405	9410	9411
	9412	9516	9517	9537	9538	9665	9670	9671	9672	9785	9786	9810	9811	9848	9853
	9854	9855	9914	9964	9969	9970	9971	10016	10028	10035	10047	10100	10105	10106	10107
	10181	10214	10215	10216	10217	10226	10227	10228	10274	10277	10281	10287	10290	10305	11100
	11110	11117	11143	11146	11147	11150	11176	11180	11247	11317	11394	11409	11419	11480	11483
	11491	11503	11504	11512	11517	11533	11545	11598	11600	11618	11641	11642	11880	11957	11963
	11995	12011	12028												
.IFT	1582	1586	1591	1595	1599	1604	1608	1663	1705	1709	1768	1795	1799	1803	1807
	1854	1860	1866	1938	1942	1980	1986	1992	2025	2031	2037	2067	2071	2075	2081
	2218	2224	2250	2256	2338	2468	2639	2645	2665	2671	2692	2705	2716	2723	2763
	2774	2783	2790	2797	2915	2919	2933	2939	2943	2970	2976	3012	3016	3116	3118
	3120	3121	3123	3125	3127	3129	3131	3133	3135	3157	3212	3215	3269	3271	3273
	3274	3276	3278	3280	3282	3284	3286	3288	3310	3353	3364	3374	3429	3513	3524
	3546	3548	3550	3551	3553	3555	3557	3559	3561	3563	3565	3587	3652	3663	3739
	3741	3743	3744	3746	3748	3750	3752	3754	3756	3758	3759	3824	3835	3943	3945
	3947	3948	3950	3952	3954	3956	3958	3960	3962	3984	4063	4165	4167	4169	4170
	4172	4174	4176	4178	4180	4182	4184	4185	4251	4255	4262	4347	4351	4395	4399
	4403	4404	4406	4410	4414	4418	4422	4426	4430	4434	4458	4542	4564	4574	4582
	4608	4614	4694	4697	4700	4701	4702	4705	4708	4711	4717	4719	4721	4743	4817
	4824	4831	4937	4944	4951	5137	5139	5142	5143	5144	5145	5286	5291	5292	5293
	5294	5449	5453	5454	5455	5456	5643	5647	5648	5649	5650	5808	5812	5813	5814
	5815	5995	5999	6000	6001	6002	6149	6153	6154	6155	6156	6344	6348	6349	6350
	6351	6509	6513	6514	6515	6516	6720	6724	6725	6726	6727	6910	6818	6910	6914
	6915	6916	6917	7154	7156	7159	7160	7161	7162	7320	7325	7326	7327	7328	7567
	7569	7572	7573	7574	7575	7738	7743	7744	7745	7746	7985	7987	7990	7991	7992

DZRJIA.P11 CROSS REFERENCE TABLE

```
             7993   8148   8153   8154   8155   8156   8376   8380   8387   8451   8456   8457   8458   8459   8629
             8632   8639   8709   8714   8715   8716   8717   8942   8944   8947   8948   8949   8950   9037   9042
             9043   9044   9045   9155   9157   9160   9161   9162   9163   9405   9409   9410   9411   9412   9665
             9669   9670   9671   9672   9848   9852   9853   9854   9855   9903   9909   9914   9964   9968   9969
             9970   9971  10016  10028  10035  10044  10100  10104  10105  10106  10107  10170  10176  10181  10233
            10239  10887  10893  10900  10904  10908  10915  10919  11045  11051  11063  11069  11075  11079  11085
            11091  11095  11105  11158  11482  11487  11568  11588  11595  11628  11673  11681  11689  11697  11705
            11713  11721  11729  11737  11745  11753  11761  11769  11777  11785  11793  11801  11809  11817
.IFTF        1582   1586   1591   1595   1599   1604   1608   1663   1705   1709   1768   1795   1799   1803   1807
             1854   1860   1866   1938   1942   1980   1986   1992   2025   2031   2037   2067   2071   2075   2081
             2218   2224   2250   2256   2639   2645   2665   2671   2915   2919   2933   2939   2943   2970   2976
             3012   3016   3513   3524   4347   4351   4542   4608   4614   6810   6818   9903   9909  10170  10176
            10233  10239  10887  10893  10900  10904  10908  10915  10919  11045  11051  11063  11069  11075  11079
            11085  11091  11095  11105  11156  11419  11480  11483  11564  11572  11594  11627  11673  11681  11689
            11697  11705  11713  11721  11729  11737  11745  11753  11761  11769  11777  11785  11793  11801  11809
            11817
.IIF            2      7     12     44     45     46     48     51     52     53     54     55     56    174    302
             1532   1535   1541   1542   1544   1545   1988   1994   2301   2302   2325   2326   2406   2407   2430
             2431   2838   2839   2996   2997   3051   3052   3075   3076   3166   3167   3190   3191   3326   3327
             3592   3618   3631   3764   3790   3803   3986   3989   4010   4011   4188   4189   4210   4211   4371
             4372   4462   4463   4486   4487   4632   4633   4746   4769   4770   4865   4866   4889   4890   5020
             5021   5043   5044   5147   5148   5177   5178   5296   5297   5326   5327   5458   5459   5488   5489
             5652   5653   5682   5683   5817   5818   5847   5848   6004   6005   6034   6035   6158   6159   6188
             6189   6353   6354   6383   6384   6518   6519   6548   6549   6729   6730   6760   6761   6919   6920
             6950   6951   7164   7165   7193   7194   7330   7331   7360   7361   7577   7578   7607   7608   7748
             7749   7778   7779   7922   7923   7995   7996   8025   8026   8158   8159   8189   8190   8332   8333
             8461   8462   8482   8483   8546   8547   8584   8585   8719   8720   8740   8741   8828   8829   8952
             8953   8982   8983   9047   9048   9077   9078   9165   9166   9196   9197   9320   9321   9414   9415
             9443   9444   9581   9582   9674   9675   9708   9709   9857   9858   9878   9879   9973   9974   9996
             9997  10109  10110  10124  10125  10275  10281  10282  10293  10305  10309  11120  11121  11122  11123
            11124  11125  11129  11157  11158  11173  11176  11177  11314  11317  11323  11362  11363  11422  11526
            11534  11541  11595  11601  11602  11603  11604  11605  11610  11633  11641  11642  11978  11979  11980
            11981  11982  11984  11986  11987  11988  11989  11990
.IRP         1524   1675   1721   1744   1890   2051   2083   2124   2164   2377   2516   2535   2552   2854   3028
             3116   3125   3137   3153   3157   3245   3269   3278   3290   3306   3310   3340   3386   3460   3546
             3555   3567   3583   3587   3701   3739   3748   3759   3882   3901   3920   3943   3952   3964   3980
             3984   4104   4123   4141   4165   4178   4287   4394   4417   4436   4452   4645   4693   4717   4723
             4739   4990   5057   5414   5571   5933   6272   6635   7042   7451   7866   8512   8772   8887   8904
             9268   9367   9384   9515   9628   9645   9784  10215  10324  10360  10376  10386  10405  10414  10433
            10439  10552  10561  10614  10674  10697  10740  10752  10764  10782  10795  10819  10851  10939  10948
            11002  11030  11190  11230  11558  11584  11611  11998  12004  12017  12018
.LIST           2     13     26     40     51    167    174    282    284    285    286    287    288    289    290
              291    292    293    294    295    296   1434   1524   1546   1564   1582   1586   1591   1595   1599
             1604   1608   1663   1675   1682   1705   1709   1721   1728   1744   1758   1769   1795   1799   1803
             1807   1854   1860   1866   1890   1904   1906   1938   1942   1980   1986   1992   2025   2031   2037
             2051   2060   2067   2071   2075   2081   2083   2092   2124   2131   2164   2192   2218   2224   2250
             2256   2330   2335   2336   2377   2387   2394   2437   2442   2443   2516   2520   2535   2539   2552
             2571   2584   2639   2645   2665   2671   2844   2849   2850   2854   2864   2898   2915   2919   2933
             2939   2943   2970   2976   3002   3007   3008   3012   3016   3028   3037   3080   3085   3096   3195
             3200   3201   3245   3253   3331   3336   3337   3460   3489   3513   3524   3636   3641   3642   3701
             3723   3808   3813   3814   3882   3886   3901   3905   3920   3930   4015   4020   4021   4104   4108
             4123   4127   4141   4151   4215   4220   4221   4287   4309   4324   4347   4351   4376   4381   4382
             4542   4608   4614   4638   4643   4644   4645   4665   4666   4775   4780   4781   4895   4900   4901
             4990   4997   4999   5023   5028   5029   5046   5051   5052   5057   5084   5185   5190   5191   5332
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
          5337  5338  5414  5426  5494  5499  5500  5571  5600  5690  5695  5696  5853  5858  5859
          5933  5962  6042  6047  6048  6194  6199  6200  6272  6301  6391  6396  6397  6554  6559
          6560  6635  6667  6668  6768  6773  6774  6810  6818  6956  6961  6962  7042  7073  7201
          7206  7207  7368  7373  7374  7451  7486  7615  7620  7621  7784  7789  7790  7866  7885
          7927  7932  7933  8035  8040  8041  8195  8200  8201  8339  8344  8345  8486  8491  8492
          8512  8525  8552  8557  8558  8591  8596  8597  8744  8749  8750  8772  8798  8834  8839
          8840  8986  8991  8992  9081  9086  9087  9202  9207  9208  9268  9289  9324  9329  9330
          9449  9454  9455  9515  9538  9585  9590  9591  9714  9719  9720  9784  9811  9826  9883
          9888  9889  9903  9909 10000 10005 10006 10128 10133 10134 10170 10176 10215 10227 10233
         10239 10281 10297 10687 10893 10900 10904 10908 10915 10919 11045 11051 11063 11069 11075
         11079 11085 11091 11095 11105 11124 11503 11641 11673 11681 11689 11697 11705 11713 11721
         11729 11737 11745 11753 11761 11769 11777 11785 11793 11801 11809 11817 11970 11978 11979
         11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
.MACRO      40    52   245  1674  1720  1743  1889  2050  2082  2123  2164  2376  2551  2853  3027
          3245  3459  3699  3918  4139  4287  4645  4990  5057  5413  5570  5933  6272  6635  7041
          7450  7865  8511  8772  9267  9514  9783 10215 11970
.MCALL       2   167  1546
.NLIST       3    13    26    40    51   167   174   282   284   285   286   287   288   289   290
           291   292   293   294   295   296  1434  1524  1546  1564  1582  1586  1591  1595  1599
          1604  1608  1663  1675  1682  1705  1709  1721  1728  1744  1758  1768  1795  1799  1803
          1807  1854  1860  1866  1890  1904  1906  1938  1942  1980  1986  1992  2025  2031  2037
          2051  2060  2067  2071  2075  2081  2083  2092  2124  2131  2164  2192  2218  2224  2250
          2256  2330  2335  2336  2377  2387  2394  2437  2442  2443  2516  2520  2535  2539  2552
          2571  2594  2639  2645  2665  2671  2844  2849  2850  2854  2864  2898  2915  2919  2933
          2939  2943  2970  2976  3002  3007  3008  3012  3016  3028  3037  3080  3085  3086  3195
          3200  3201  3245  3253  3331  3336  3337  3460  3489  3513  3524  3636  3641  3642  3701
          3723  3808  3813  3814  3882  3886  3901  3905  3920  3930  4015  4020  4021  4104  4108
          4123  4127  4141  4151  4215  4220  4221  4287  4309  4324  4347  4351  4376  4381  4382
          4542  4608  4614  4638  4643  4644  4645  4665  4666  4775  4780  4781  4895  4900  4901
          4990  4997  4999  5023  5028  5029  5046  5051  5052  5057  5084  5185  5190  5191  5332
          5337  5338  5414  5426  5494  5499  5500  5571  5600  5690  5695  5696  5853  5858  5859
          5933  5962  6042  6047  6048  6194  6199  6200  6272  6301  6391  6396  6397  6554  6559
          6560  6635  6667  6668  6768  6773  6774  6810  6818  6956  6961  6962  7042  7073  7201
          7206  7207  7368  7373  7374  7451  7486  7615  7620  7621  7784  7789  7790  7866  7885
          7927  7932  7933  8035  8040  8041  8195  8200  8201  8339  8344  8345  8486  8491  8492
          8512  8525  8552  8557  8558  8591  8596  8597  8744  8749  8750  8772  8798  8834  8839
          8840  8986  8991  8992  9081  9086  9087  9202  9207  9208  9268  9289  9324  9329  9330
          9449  9454  9455  9515  9538  9585  9590  9591  9714  9719  9720  9784  9811  9826  9883
          9888  9889  9903  9909 10000 10005 10006 10128 10133 10134 10170 10176 10215 10227 10233
         10239 10281 10297 10887 10893 10900 10904 10908 10915 10919 11045 11051 11063 11069 11075
         11079 11085 11091 11095 11105 11124 11503 11641 11673 11681 11689 11697 11705 11713 11721
         11729 11737 11745 11753 11761 11769 11777 11785 11793 11801 11809 11817 11970 11978 11979
         11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
.PAGE       53    57   168   208   245   303 10445 10473 10493 10540 10743 10858 11114 11177 11244
         11314 11542 11595 11954 11992
.RADIX    2335  2336  2442  2443  2849  2850  3007  3008  3085  3086  3200  3201  3336  3337  3641
          3642  3813  3814  4020  4021  4220  4221  4381  4382  4643  4644  4790  4781  4900  4901
          5028  5029  5051  5052  5190  5191  5337  5338  5499  5500  5695  5696  5858  5859  6047
          6048  6199  6200  6396  6397  6559  6560  6773  6774  6961  6962  7206  7207  7373  7374
          7620  7621  7789  7790  7932  7933  8040  8041  8200  8201  8344  8345  8491  8492  9557
          8558  8596  8597  8749  8750  8839  8840  8991  8992  9086  9087  9207  9208  9329  9330
          9454  9455  9590  9591  9719  9720  9888  9889 10005 10006 10133 10134
.REPT       14    22   174   284   290
.SBTTL      40    57   168   178   189   208   245   303  1322  1505  1506  1507  1510.  1525  1612
```

DZRJIA.P11 CROSS REFERENCE TABLE

```
          1675    1721    1744    1890    2051    2083    2124    2164    2377    2516    2535    2552    2854    3019    3020
          3021    3028    3245    3460    3701    3882    3901    3920    4104    4123    4141    4287    4645    4990    5056
          5057    5414    5571    5933    6272    6635    7042    7451    7454    7866    8512    8770    8772    9268    9515
          9784   10215   10266   10267   10268   10271   10312   11114   11177   11244   11314   11542   11595   11646   11877
         11954   11970   11992
.TITLE       2
.WORD      174     175     176     186     253     256     257     258     259     262     263     264     265     266     267
           268     271     272     273     282     284     285     286     287     288     289     290     291     292     293
           294     295    1428    1473    1474    1476    1480    1481    1502    1503    1504   10286   10289   10304   10857
         11311   11318   11319   11320   11591   11594   11846   11851   11953   12027   12029   13571   13574   13577   13579
         13581   13584   13587   13590   13593   13597   13600   13602   13604   13606   13610
```

ERRORS DETECTED: 0

```
*DZRJIA,DZRJIA/SOL/CRF←DZRJIA
RUN-TIME:  151  154  19 SECONDS
CORE USED:  29K
```