

RP04/05/06

**FUNCTION CONTROL PART 2
MD-11-DZRJJ-A**

EP-DZRJJ-A-DL-A

NOV 1976

COPYRIGHT©1976

FICHE 1 OF 2 MADE IN USA

RP04/05/06

functional cont. TEST 2
MD-11-DZRJJ-A

EP-DZRJJ-A-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

NOV 1976
digital
MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRJJ-A-D
 PRODUCT NAME: RPO4 5/6 FUNCTIONAL CONTROLLER TEST PART II
 DATE CREATED: MAY 1975
 MAINTAINER: DIAGNOSTIC ENGINEERING
 AUTHOR: FETE BLACKSTONE

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT 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.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO LIABILITY FOR THE ACCURACY OR RELIABILITY OF ITS INFORMATION. THE INFORMATION IS NOT SUPPLIED BY DIGITAL.

CONTENTS

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

1.2 ABSTRACT

THIS DIAGNOSTIC TESTS THE DCL OF THE RPO4/5/6 DISK SUBSYSTEM WHEN CONNECTED TO EITHER AN RH11 OR RH70 CONTROLLER.

IT USES THE DISK SURFACE AND THE DRIVE MECHANICS TO PROVE THE PROPER WORKING OF THE SUBSYSTEM. IT DOES NOT NEED A FORMATTED DISK PACK. A DISK PACK WITH NO VITAL INFORMATION WRITTEN ON IT IS ESSENTIAL. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC IT CAN BE ASSERTED THAT THE DCL IN THE RPO4/5/6 SUBSYSTEM WORKS SUCCESSFULLY WHILE STANDING ALONE. SYSTEMS INTERACTION AND DRIVE TIMING IS LEFT TO OTHER DIAGNOSTICS. THIS IS WITH THE ASSUMPTION THAT STATIC 1 (DZRP5 AND DZRP7) HAS BEEN RUN SUCCESSFULLY.

2.0 REQUIREMENTS

2.1 EQUIPMENT

RDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RPO4/5/6 DISK SYSTEM. THE RPO4/5/6 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-DZRIJ-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.

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

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

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

SWITCH 12 MUST BE SET WHEN THIS PROGRAM IS TO BE RUN USING AN RHP/C 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

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RPO4/5/6S ON THE SYSTEM WILL BE TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT. TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER THAT RESPONDS) THEN GO ON TO THE NEXT HIGHER UNIT NUMBER THAT IS POWERED UP.

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

4.3 PROGRAM AND/OR OPERATOR ACTION

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I. E. AN 11/34) IT WILL DETERMINE THAT A HARDWARE SWITCH REGISTER IS NOT PRESENT, AND WILL USE A "SOFTWARE" SWITCH REGISTER. THE SETTINGS OF THE "SOFTWARE" SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE A 'CONTROL G' AT ANY TIME EXCEPT WHEN IT IS AT A HIGHER PRIORITY PROCESSING AN RPO4/S/6 INTERRUPT. THE "SOFTWARE" SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO PROMPTING FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

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

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

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

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

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

SWITCH 12 - RH70 CONTROLLER SELECT
THIS SWITCH MUST BE SET AT THE START OF THE PROGRAM WHEN THE DISK DRIVES TO TESTED ARE CONNECTED TO AN RH70 CONTROLLER.

IT MUST NOT BE SET WHEN DISK DRIVES TO BE TESTED ARE CONNECTED TO AN RH11 CONTROLLER.

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

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

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

NOTE: SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR (7:0)
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7 HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH 0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPRIATE 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 MEANING 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 SUBSEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY

THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER EXAMPLE OF SWITCH 9 BEING LOW IS WITH SWITCH 6, WHICH IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH 9 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 RPO4/5/6 REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS CANNOT BE CLEARED PRIOR TO THE START OF A TEST SEQUENCE - THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN ADDITION, THE TTY BELL WILL RING AND THE PROGRAM WILL HALT.

IT IS SUGGESTED THAT IF THIS HAPPENS, THE OPERATOR LOAD ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT TO SOLVE THE PROBLEM. IF THE FAILURE CONTINUES TO OCCUR, THERE ARE TWO OPTIONS FOR THE OPERATOR:

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

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

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

7.0 RESTRICTIONS

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

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE APPROXIMATELY 20 SECONDS. SUBSEQUENT PASSES WILL TAKE 60 SECONDS.

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS. FOR INSTRUCTIONS REGARDING THE USAGE OF THIS TECHNIQUE, HIT ↑ 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

CO, TO, SO

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

CO, TO, S1

TT27-W,R, TT37-W,R, TT40-R, TT41-W,R, TT42-W,R, TT43-W,R

CO, T18, S21

TT30-W, TT31-W,R

C1, TO, SO

TT30-W,R, TT31-W,R, TT53-W,R, TT54-W,R

C1, T18, S21

TT31-?

C2, TO, SO
TT31-W,R

C2, T18, S21
TT31-W

C3, TO, SO
TT31-W,R

C3, T18, S21
TT31-W

C4, TO, SO
TT31-W,R

C4, T18, S21
TT31-W

C5, TC, SO
TT31-W,R

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

C5, T18, S21
TT31-W

C6, TO, SO
TT31-W,R

C6, T18, S21
TT31-W

C7, TO, SO
TT31-W,R

C7, T18, S18
TT31-W

C8, TO, SO
TT31-W,R

C8, T18, S21
TT31-W

C9, TO, SO
TT31-W

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

C10, TO, SO
TT31-W,R

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

9.2

THE FOLLOWING SECTION DESCRIBES EACH TEST AND SUBROUTINES
IN DETAIL AND CAN BE USED AS AN INDEX TO THE LISTING.

MO1

THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING
WHERE THAT ITEM WILL BE FOUND.

SEQ 0011

DOCUMENT

MNDEC-11-DZRJJ-A, RPO4/5/6 FUNCT. CONT. TST-PT 2

COPYRIGHT 1976
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

TABLE OF CONTENTS

46 OPERATIONAL SWITCH SETTINGS

61 BASIC DEFINITIONS

172 TRAP CATCHER

182 ACT11 HOOKS

193 STARTING ADDRESS

210 MEMORY MANAGEMENT DEFINITIONS

247 COMMON TAGS

305 ERROR POINTER TABLE

1239 REGISTER ADDRESSES

1412

1413 *** DIAGNOSTIC CODE ***

1414

1417 SETUP TESTS

1432 INITIALIZE THE COMMON TAGS

1518 GET VALUE FOR SOFTWARE SWITCH REGISTER

2199 DATA TRANSFER RELATED ERRORS (USING MEDIA)

8971

8972 ***SLE ROUTINES***

8973

8978 END OF PASS ROUTINE

9775 SYSMAC LIBRARY ROUTINES

9778 SCOPE HANDLER ROUTINE

TABLE OF CONTENTS
*****+*****

9841 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9908 TYPE ROUTINE
9978 TTY INPUT ROUTINE
10206 READ AN OCTAL NUMBER FROM THE TTY
10259 ERROR HANDLER ROUTINE
10310 ERROR MESSAGE TYPEOUT ROUTINE
10541 BINARY TO OCTAL (ASCII) AND TYPE
10618 TRAP DECODER
10634 TRAP TABLE
10655 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-DZGAC-00), MAR 21, 1976.

33 INTERNAL PROGRAM MACROS BEGIN HERE

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

46 *****
OPERATIONAL SWITCH SETTINGS

	SWITCH	USE
	15	HALT ON ERROR
	14	LOOP ON TEST
	13	INHIBIT ERROR TYPEOUTS
	12	RH70 CONTROLLER SELECT
	11	INHIBIT ITERATIONS
55	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

61 *****
BASIC DEFINITIONS

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

68 MISCELLANEOUS DEFINITIONS

80 GENERAL PURPOSE REGISTER DEFINITIONS

92 PRIORITY LEVEL DEFINITIONS

102 "SWITCH REGISTER" SWITCH DEFINITIONS

130 DATA BIT DEFINITIONS (BIT00 TO BIT15)

158 BASIC "CPU" TRAP VECTOR ADDRESSES

172

TRAP CATCHER

175 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

182

ACT11 HOOKS

193

STARTING ADDRESS

202 STARTING ADDRESS 200 FOR NORMAL STARTS
THIS WILL TEST ALL RPO4'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
AT THE DRIVE

210

MEMORY MANAGEMENT DEFINITIONS

212 KT11 VECTOR ADDRESS

216 KT11 STATUS REGISTER ADDRESSES

223 KERNEL "I" PAGE DESCRIPTOR REGISTERS

234 KERNEL "I" PAGE ADDRESS REGISTERS

1
R
2
1

247 *****
COMMON TAGS

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

305 *****
ERROR POINTER TABLE

307 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:

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

1239 *****
REGISTER ADDRESSES

1319 TABLE OF FUNCTIONS FOR RHCSI THEN "GO" BIT HAS TO BE SET

1342 DATA BUFFER FOR READ WRITE

1412 *****

1413 *****
*** DIAGNOSTIC CODE ***

1414 *****

```

1417 *****
      SETUP TESTS
      *****

1432 *****
      INITIALIZE THE COMMON TAGS
      *****

1518 *****
      GET VALUE FOR SOFTWARE SWITCH REGISTER
      *****

1533  IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM ?
      IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
      IF SO MAKE RPO4 INTERRUPTS GO TO 'TIME 1'
      IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
      IF NOT MAKE RPO4 INTERRUPTS GO TO 'TIME 2'

1539  THE NEXT LINE IS TO BE ADDED LATER
      AND THE JUMP AND NOP REMOVED
      FOR NOW NO CLOCK WILL BE USED

1543  MOV    2#15,2#ERRVEC ;SET TIME-OUT VECTOR

1579  TEST 1 REFERENCE EACH REGISTER

1581          REFERENCE EACH REGISTER BY A MOVE INSTRUCTION

1627  TEST 2 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT

1629          CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES

1650  TEST 3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

1652          THE NUMBER OF RPO4/RPO6 DRIVES PRESENT IS FOUND
          BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
          IN RHCS2 INCREMENTED FROM ZERO TO SEVEN.

1656          THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT.

1658          THE DRIVE TYPE IS CHECKED TO BE RPO4 OR RPO6 AND
          UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'

1719  SET UP THE UNITS TABLE

1795  TEST 4 TYPE SERIAL NUMBER AND DRIVE TYPE

1797          SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
          TYPE UNIT UNDER TEST

```

Handwritten mark resembling '11/11'

Handwritten mark resembling a circled '11'

1800 READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
TYPE THEM OUT AND PROCEED

1803 TO LOOP HERE SET SWITCH 9, AND THIS TEST NUMBER ON
SWITCHES 0 THRU 7 AND RESTART

1829 INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
& DECREMENT "NCUNITS" PRESENT TO BE TESTED

1893 TYPE OUT THE DRIVE TYPE IN ASCII

1940 TEST 5 CHECK MOL TO BE HIGH

1942 MAKE SURE THAT DRIVE IS ON LINE BEFORE STARTING PROGRAM
IF DRIVE IS OFF LINE THEN AFTER TYPE OUT THE PROGRAM WILL
HANG FOR EVER WAITING FOR DRIVE TO GO ON LINE

1992 TEST 6 PROGRAM INTERRUPT

1984 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

2029 TEST 7 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

2031 PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
IE AND RDY IS SET. THIS SHOULD NOT INTERRUPT

2072 TEST 10 SET VV BIT #6 IN RHDS1

2074 THIS TEST SETS VV IN RHDS1
THERE IS A RESET AT THE BEGINING OF THE TEST
FOR ERROR RECOVERY ONLY.

2081 IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN

2102 NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE

2137 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED

2168 NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
WITH AFTER PACK ACKNOWLEDGE

2199

DATA TRANSFER RELATED ERRORS (USING MEDIA)

2203 TEST 11 LAST BLOCK TRANSFERED-RHDS1 LBT

2205 WRITE ONE WORD OF 15125 ON CYLINDER 410./814., TRACK 18
SECTOR 21, BY A WRITE HEADER AND DATA COMMAND
THEN CHECK ALL REGISTERS (LAST BLOCK TRANSFERED
SHOULD BE SET)

2210 THEN READ ABOVE USING READ DATA (256 WORDS)
CHECK ALL REGISTERS AND DATA
(AGAIN 'LBT' SHOULD BE SET)

2224 CHECK THE DRIVE TYPE AND THEN FILL THE
WRITE FROM BUFFER WITH APPROPRIATE HEADER

2255 FILL READ INTO BUFFER WITH ALL ONES

2265 DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
WRITE HEADER AND DATA COMMAND IS LOADED

2313 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE

2340 TIME IS NOT CRITICAL HERE

2349 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

2380 COMPARE ALL REGISTERS

2402 NOW A READ DATA WILL BE DONE ON SAME CYLINDER, SECTOR 3 TRACK

2404 CLEAR ANY PREVIOUS ERRORS

2412 FILL WRITE FROM BUFFER WITH EXPECTED DATA

2425 FIRST THE DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
READ COMMAND IS LOADED

2470 SAVE REGISTERS FOR COMPARISON AFTER READ DATA

2495 TIME IS NOT CRITICAL HERE

2504 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

2536 COMPARE ALL REGISTERS

2556 READ DATA WILL BE COMPARED

2576 TEST 12 SEARCH COMMAND

2578 THE SEARCH COMMAND WILL BE DONE ON CYLINDER 0
THAT IS STARTING WITH A RECALIBRATE
THEN HEADER AND DATA WILL BE WRITTEN FOR SECTOR 0 AND 1
ALL REGISTERS WILL BE CHECKED
A SEARCH COMMAND WILL BE GIVEN FOR SECTOR 0
ON INTERRUPT SECTOR 1 HEADER AND DATA WILL BE READ
TIME WILL BE CRITICAL AS THE TIME TAKEN TO DO THE
READ IS THE ONLY INDICATOR THAT THE HEADS WERE ON
SECTOR 0 AT INTERRUPT TIME. TIME ALLOWED IS MAXIMUM
OF 1500 MICRO SECONDS
THEN ALL REGISTERS ARE CHECKED AND DATA READ
IS CHECKED

2600 GET HEADS TO CYLINDER 0

2634 FILL WRITE FROM BUFFER WITH HEADER

2644 FILL WRITE FROM BUFFER WITH DATA

2652 FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER

2662 FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA

2670 CLEAR READ INTO BUFFER WITH DATA OTHER THAN EXPECTED DATA

2678 THE WRITE HEADER AND DATA WILL BE LOADED

2697 SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

2727 ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC

2736 COMPARE CONTENTS OF RHCSI AND RHDSI ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED

2750 NOW CHANGE SAVE REGISTERS TO EXPECTED VALUES

2777 NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
WITH REGISTERS AFTER COMMAND

2797 NOW A SEARCH COMMAND WILL BE GIVEN
BUT BEFORE THAT ALL POSSIBLE REGISTERS
WILL BE FILLED FOR THE READ HEADER AND DATA SECTOR 1
AS THERE WILL NOT BE MUCH TIME BETWEEN THE
COMPLETION OF THE SEARCH AND THE SECTOR 1 COMING.

2803 FILL FOR THE READ HEADER AND DATA COMMAND WHICH WILL NOT
BE EXECUTED TILL AFTER THE SEARCH
THE SEARCH WILL ONLY LEAVE RHCSI AND RHDSI
CHANGED ALL THE REST WILL BE UNCHANGED

Ex. 1/21

2831 SAVE REGISTERS FOR COMPARISON NOT AFTER THE
SEARCH COMMAND BUT AFTER THE READ HEADER AND DATA

2840 NOW SAVE VALUES FOR RHCS1 AND RHDST WHICH
WILL CHANGE AFTER THE SEARCH

2848 THE INTERRUPT VECTOR WILL BE SET TO GO TO 25
AFTER THE SEARCH

2863 TIME IS NOT CRITICAL THIS ONLY WAITS FOR SEARCH COMPLETION

2899 TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
INDICATES WRONG SEARCH IN THE SEARCH COMMAND

2999 WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA

2915 CHANGE SAVED REGISTERS TO EXPECTED VALUES

2937 COMPARE REGISTER BEFORE READ HEADER AND DATA
WITH REGISTERS AFTER COMMAND

2956 NOW READ INTO BUFFER WILL BE CHECKED TO SEE
THE READ WAS GOOD

2975 TEST 13 SEARCH COMMAND

2977 THE ONLY THING NEW IN THIS TEST IS AN IMPLIED SEEK
IN A SEARCH COMMAND
THE HEADS START FROM CYLINDER 10 BY A SEEK
COMMAND THEN A SEARCH SECTOR 0 TRACK 0 CYLINDER 0
IS GIVEN
THEN A READ COMMAND IS GIVEN FOR
CYLINDER 0, TRACK 0, SECTOR 1
TIME FOR THE READ IS THE ONLY INDICATOR
OF CORRECT SEARCH

2997 GET THE HEADS TO CYLINDER 10

3027 FILL REGISTERS FOR READ HEADER AND DATA TO BE DONE AFTER SEARCH

3050 SAVE REGISTERS FOR COMPARISON AFTER SEARCH
AND READ HEADER AND DATA

3059 NOW GIVE THE SEARCH COMMAND

3111 THIS IS ONLY A WAIT LOOP

3135 TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
INDICATES WRONG SEARCH IN THE SEARCH COMMAND

3145 WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA

3161 CHANGE SAVED REGISTERS TO EXPECTED VALUES

3188 COMPARE REGISTER BEFORE READ HEADER AND DATA

3199 WITH REGISTERS AFTER COMMAND

3208 NOW READ INTO BUFFER WILL BE CHECKED TO SEE
THE READ WAS GOOD

3230 THE NEXT TEST REMOVES SECTOR 1 ON CYLINDER 0
TRACK0 AND PUTS SECTOR C THERE.
HENCE THE PACK IS UNFORMATTED FROM
THIS POINT ON TO THE TEST WHEN SECTOR
1 IS REPLACED. IF TESTING IS STOPPED WITH
AN ERROR IN THE SECTION OF THE PROGRAM BETWEEN
THIS AND WHEN SECTOR 1 IS REPLACED THEN THE
DISK BEING USED MAY HAVE BEEN UNFORMATTED
IF THE LAST PASS OF THIS PROGRAM GIVES
NO ERRORS IN THIS SECTION THEN THE DISK
MAY NOT HAVE BEEN UNFORMATTED. HOWEVER IT
IS RECOMMENDED THAT AFTER A PASS OF THIS
PROGRAM THE DISK BE REFORMATTED.

3251 TEST 14 HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)

3253 WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
AND PUT SECTOR 0 THERE ON CYLINDER 0
THEN A READ DATA IS GIVEN FOR SECTOR1
HCE- BIT #7 IN RHER1 SHOULD SET.
ALL REGISTERS ARE CHECKED
ANY DATA READ IS CHECKED

3270 FILL WRITE FROM BUFFER WITH HEADER

3281 FILL READ INTO BUFFER WITH ALL ONES

3290 WRITE HEADER AND DATA IS LOADED

3309 NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE

3338 TIME IS NOT CRITICAL

3347 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

3364 COMPARE ALL REGISTERS

3384 NOW A READ DATA WILL BE DONE ON CYLINDER=0, SECTOR=1.
TRACK=0
FILL WRITE FROM BUFFER WITH EXPECTED DATA

P
K

3401 READ COMMAND IS LOADED
3419 SAVE REGISTERS FOR COMPARISON AFTER READ DATA
3447 TIME IS NOT CRITICAL
3456 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3492 COMPARE ALL REGISTERS
3510 READ DATA WILL BE COMPARED
3526 TEST 15 HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)
3528 WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
AND PUT SECTOR 0 ON CYLINDER 0
THEN A WRITE DATA IS GIVEN FOR SECTOR 1, TRACK 0, CYLINDER 0
FOR 70. WORDS
HCE - BIT 7 IN RHER1 SHOULD SET
ALL REGISTERS ARE CHECKED
THEN A READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
IS GIVEN. HCE - BIT 7 SHOULD SET AND ALL
HEADER AND DATA SHOULD BE READ
3548 FILL WRITE FROM BUFFER WITH HEADER AND DATA
3560 FILL READ INTO BUFFER WITH ALL ONES
3568 WRITE HEADER AND DATA IS LOADED
3587 NOW SAVE REGISTERS FOR COMPARISON AFTER
WRITE HEADER AND DATA
3617 TIME IS NOT CRITICAL
3626 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3643 COMPARE ALL REGISTERS
3661 NOW A WRITE DATA WILL BE DONE ON SAME CYLINDER, SECTOR
TRACK
3665 FILL WRITE FROM BUFFER WITH DATA
3673 WRITE DATA COMMAND IS LOADED
3691 SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
3719 TIME IS NOT CRITICAL

3728 NOW CHANGE REGISTERS TO EXPECTED VALUE

3809 COMPARE ALL REGISTERS

3828 READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
WILL BE ATTEMPTED

3837 FILL WRITE FROM BUFFER WITH EXPECTED DATA

3861 FILL READ INTO BUFFER WITH ALL ONES

3869 FILL REGISTERS WITH READ HEADER AND DATA COMMAND

3887 SAVE REGISTERS FOR COMPARISON AFTER READ
HEADER AND DATA

3916 TIME IS NOT CRITICAL

3925 CHANGE SAVED REGISTERS TO EXPECTED VALUE

3969 COMPARE REGISTERS BEFORE READ HEADER AND DATA
WITH AFTER

3989 NOW COMPARE READ DATA
THE COMMAND READ ONLY 204 WORDS, 4 HEADER WORDS
AND 200 DATA WORDS

4020 TEST 16 HEADER COMPARE ERROR - RHER1 HCE

4022 WITH THE HEADS ON CYLINDER 0 A SEARCH COMMAND IS GIVEN
FOR CYLINDER 0 TRACK 0 SECTOR 1, ALTHOUGH THE HEADER
FOR THIS SECTOR IS CHANGED TO SECTOR 0 HCE-BIT #7
IN RHER1 SHOULD NOT SET
BECAUSE SEARCH DOES NOT READ HEADER.BUT ONLY USES SECTOR COUNTER

4038 GET HEADS TO CYLINDER 0

4075 FILL REGISTERS FOR SEARCH

4083 SAVE REGISTERS FOR COMPARISON AFTER SEARCH

4121 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED

4145 CHANGE SAVED REGISTERS TO EXPECTED VALUE

4165 COMPARE REGISTERS BEFORE SEARCH WITH AFTER SEARCH

4188 TEST 17 RESTORE SECTOR 1 CYLINDER 1 TRACK 1

4190

THIS REPLACES REMOVED SECTOR

WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 1, KEYS=0, NUMBER OF WORDS 256 WORDS
 OF 0

THEN READ HEADER AND DATA FOR ABOVE.
 WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
 10000100 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

4216

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

4229

FILL WRITE FROM BUFFER WITH HEADER

4239

FILL WRITE FROM BUFFER WITH DATA

4247

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

4257

NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

4286

NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

4315

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

4324

COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 RD AND RS IMMEDIATELY AFTER GO IS ISSUED

4348

NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

4365

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

4396 NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
NOTHER GOT CHANGED

4401 NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
READ INTO BUFFER IS FILLED WITH ONES

4416 NOW FILL COMMAND

4435 NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA

4473 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
RC AND RS IMMEDIATELY AFTER GO IS ISSUED

4497 CHANGE SAVED REGISTERS TO EXPECTED VALUES

4514 COMPARE REGISTER BEFORE READ HEADER AND DATA
WITH REGISTERS AFTER COMMAND

4534 NOW READ INTO BUFFER WILL BE CHECKED TO SEE
THE READ WAS GOOD

4557 TEST 20 INVALID ADDRESS ERROR - RHER1 - 'IAE'

4559 A WRITE HEADER AND DATA WILL BE ATTEMPTED TO CYLINDER 411.4B15.
TRACK 0, SECTOR 0

4562 INVALID ADDRESS ERROR (IAE) BIT #10 IN RHER1 SHOULD SET

4575 CHECK THE DRIVE TYPE AND THEN FILL THE
WRITE FROM BUFFER WITH APPROPRIATE HEADER

4604 FILL WRITE FROM BUFFER WITH DATA

4613 THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE
WRITE HEADER AND DATA COMMAND IS LOADED

4663 NOW SAVE REGISTERS FOR COMPARISON AFTER
WRITE HEADER AND DATA

4703 CHANGE THE SAVED REGISTERS TO EXPECTED VALUES

4705 AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED,
RHWC, RHBA, RHCS1 & RHCS2 CANNOT BE PREDETERMINED -
THEY WILL VARY DEPENDING ON GATE DELAYS FOR DIFFERENT UNITS

4741 COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH
CONTENTS AFTER ATTEMPTED WRITE WITH AN 'IAE' ERROR

4766 TEST 21 INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE

4768 A WRITE DATA IS ATTEMPTED TO CYLINDER 0, TRACK 19,
SECTOR 0
INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
SHOULD SET

4783 FILL WRITE FROM BUFFER WITH DATA

4791 WRITE DATA COMMAND WILL BE FILLED

4809 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA

4845 CHANGE SAVED REGISTERS TO EXPECTED VALUES

4849 AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
RHWC, RHBA, RHCS1, RHCS2, CANNOT BE PEREDETERMINED
THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS

4884 COMPARE REGISTERS BEFORE ATTEMPTED WRITE DATA
WITH AFTER ATTEMPT, IAE SHOULD BE SET

4907 TEST 22 INVALID ADDRESS ERROR RHER1 -BIT #10

4909 A READ HEADER AND DATA IS ATTEMPTED TO CYLINDER 0
TRACK 0, SECTOR 22
INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
SHOULD SET
THIS WILL START WITH THE HEADS ON CYLINDER 10
TO PROVE THAT IAE SETS EVEN BEFORE THE IMPLIED
SEEK

4927 GET THE HEADS TO CYLINDER 10

4968 FILL READ INTO BUFFER WITH 125252

4976 THE READ HEADER AND DATA COMMAND IS FILLED

4995 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ

5030 CHANGE SAVED REGISTERS TO EXPECTED VALUES

5034 AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
RHWC, RHBA, RHCS1, RHCS2, CANNOT BE PEREDETERMINED
THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS

5069 COMPARE REGISTERS BEFORE ATTEMPTED READ HEADER
AND DATA WITH AFTER ATTEMPTED READ

5095 TEST 23 INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE

5097 A READ DATA IS ATTEMPTED TO CYLINDER 0, TRACK 0
SECTOR 20 - FORMAT 18 BITS PER WORD
INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
SHOULD SET

5112 FILL READ INTO BUFFER WITH 125252

5120 THE READ HEADER AND DATA COMMAND IS FILLED

5138 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ

5174 CHANGE SAVED REGISTERS TO EXPECTED VALUES

5206 COMPARE REGISTERS BEFORE ATTEMPTED READ
DATA WITH AFTER ATTEMPTED READ DATA

5232 TEST 24 ADDRESS OVERFLOW ERROR - RHER1 (BIT#9) AOE

5234 A WRITE HEADER AND DATA COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0
SECTOR 0, 256 WORDS OF 0
NO CHECK IS DONE AFTER THIS WRITE

A WRITE HEADER AND DATA COMMAND IS GIVEN FOR
CYLINDER 410./814., TRACK 18, SECTOR 21, 261 WORDS

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

A READ HEADER AND DATA CYLINDER 410./814., TRACK 18, SECTOR 21,
260+66+4=330 WORDS IS ISSUED

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

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

5265 FILL WRITE FROM BUFFER WITH HEADER

5275 FILL WRITE FROM BUFFER WITH DATA

5284 THE FIRST WRITE OPERATION IS DONE
FILL WRITE HEADER AND DATA COMMAND

5339 CHECK THE DRIVE TYPE AND DO THE
APPROPRIATE SECOND WRITE OPERATION

5342 FILL WRITE FROM BUFFER WITH HEADER
5374 FILL WRITE FROM BUFFER WITH DATA - 65125
5393 CHECK THE DRIVE TYPE AND
FILL WRITE FROM BUFFER WITH APPROPRIATE NEXT HEADER
5386 THIS IS A NON EXISTANT HEADER AND SHOULD NOT BE WRITTEN
SINCE 'AOE' SHOULD INHIBIT THE WRITE OPERATION
5417 FILL WRITE FROM BUFFER WITH DATA FOR NEXT SECTOR
5426 CHECK THE DRIVE TYPE AND DO THE APPROPRIATE
FILL WRITE HEADER AND DATA COMMAND
5474 SAVE REGISTERS FOR COMPARISON AFTER WIRTE HEADER AND DATA
5510 CHANGE SAVED REGISTERS TO EXPECTED VALUES
5560 CHECK DEVICE TYPE BEFORE SETTING UP 'RHCA' & 'RHCC'
5596 COMPARE REGISTERS BEFORE WRITE HEADER AND DATA WITH AFTER
5518 NOW PREPARE TO DO A READ HEADER AND DATA
(THE FIRST READ OPERATION)
5621 CHECK THE DRIVE TYPE AND FILL
WRITE FROM BUFFER WITH APPROPRIATE EXPECTED HEADER
5661 FILL WRITE FROM BUFFER WITH EXPECTED DATA
5669 FILL WRITE FROM BUFFER WITH 377 FROM WORDS 261 TO 266
5677 CLEAR READ INTO BUFFER
5692 CHECK THE DRIVE TYPE AND DO THE APPROPRIATE
FILL READ HEADER AND DATA COMMAND
5739 SAVE REGISTERS FOR COMPARISON AFTER
READ HEADER AND DATA
5776 CHANGE SAVED REGISTERS TO EXPECTED VALUES
5813 CHECK DRIVE TYPE BEFORE SETTING UP 'RHCA'
5841 COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
REGISTERS AFTER COMMAND
5862 NOW COMPARE THE DATA READ

5889 NOW PREPARE TO READ CYLINDER 0, SECTOR 0, TRACK 0
TO SEE THAT NOTHING GOT WRITTEN ON THERE
WITH THE ADDRESS OVER FLOW BIT SET (AOE)

5893 FILL WRITE FROM BUFFER WITH EXPECTED HEADER

5909 FILL READ INTO BUFFER WITH 377

5923 FILL COMMAND FOR READ HEADER AND DATA

5942 SAVE REGISTERS FOR COMPARISON AFTER READ

5977 CHANGE REGISTERS TO EXPECTED VALUE

6024 COMPARE DATA/READ

6049 TEST 25 FORMAT ERROR - RHER1 (BIT #4)FMT

6051 AN ATTEMPT WILL BE MADE TO WRITE DATA ON CYLINDER 0
SECTOR 0 TRACK 0 WITH 18 BITS PER WORD WHEN THE
HEADER HAS 16 BITS PER WORD SET

THIS SHOULD GIVE FORMAT ERROR FER BIT #4 IN RHER1

THEN THIS SECTOR WILL BE READ IN THE CORRECT FORMAT
16 BITS PER WORD TO CHECK THAT NOTHING GOT WRITTEN

6069 FIRST WRITE HEADER AND DATA CYLINDER 0, TRACK 0, SECTOR 0
FILL WRITE FROM BUFFER WITH HEADER

6080 FILL WRITE FROM BUFFER WITH DATA

6088 FILL COMMAND

6142 NOW PREPARE TO WRITE WITH WRONG FORMAT

6144 FILL WRITE FROM BUFFER

6152 FILL WRITE DATA COMMAND

6170 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
WITH WRONG FORMAT

6207 CHANGE SAVED REGISTERS TO EXPECTED VALUE

6286 COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT

6305 NOW PREPARE TO READ WITH CORRECT FORMAT TO CHECK
THAT NOTHING GOT WRITTEN

6313 FILL WRITE FROM BUFFER WITH EXPECTED DATA
 6321 FILL READ INTO BUFFER WITH 125252
 6329 FILL COMMAND TO READ DATA
 6347 SAVE REGISTERS FOR COMPARISON AFTER NORMAL READ
 6383 CHANGE SAVED REGISTERS TO EXPECTED VALUE
 6407 COMPARE REGISTERS BEFORE READ WITH AFTER
 6427 COMPARE DATA READ AFTER ATTEMPTED WRITE WITH
 WRONG FORMAT BIT
 6446 TEST 26 FORMAT ERROR - RHER1 (BIT #4)FMT
 6448 AN ATTEMPT IS MADE TO READ DATA WITH WRONG
 FORMAT BIT
 FORMAT ERROR BIT #4 IN RHER1 SHOULD SET
 NO DATA SHOULD BE READ
 6464 FILL WRITE FROM BUFFER WITH 107070
 6472 FILL READ INTO BUFFER WITH 107070
 6480 FILL COMMAND TO READ WITH WRONG FORMAT
 6498 SAVE REGISTERS FOR COMPARAISON AFTER READ
 6534 CHANGE SAVED REGISTERS TO EXPECTED VALUE
 6577 COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
 6596 COMPARE READ INTO BUFFER TO CHECK THAT NOTHING WAS READ
 6619 TEST 27 REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR
 6621 CYLINDER1 TRACK 0, SECTOR 0 WILL BE WRITTEN WITH
 200 WORDS OF 2000 BY A WRITE HEADER AND DATA COMMAND
 THE HEADS WILL BE BROUGHT TO CYLINDER 0 BY A SEEK
 A READ DATA COMMAND WILL BE GIVEN TO CYLINDER 1 TRACK 0
 SECTOR 0 150. WORDS. THIS WILL TAKE AT
 LEAST 7 MILI SECONDS. IMMEDIATELY AFTER GO AT
 IMPLIED SEEK TIME, WRITE INTO A REGISTER WILL BE ATTEMPTED
 THEN READY WILL BE WAITED ON TO COMPLETE THE READ DATA
 THEN ALL REGISTERS WILL BE COMPARED AND THE DATA READ
 SHOULD BE GOOD
 THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST, RHER2
 RHOF, RHCA, RHER3

6645 PREPARE TO WRITE HEADER AND DATA LINDER 1, TRACK 0, SECTOR 0
 FILL WRITE FROM BUFFER WITH HEAD

6665 FILL WRITE FROM BUFFER WITH DATA

6673 FILL COMMAND

6712 TIME IS NOT IMPORTANT

6721 NOW BRING THE HEADS TO CYLINDER 0

6756 PREPARE FOR A READ DATA

6758 FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ

6772 FILL READ INTO BUFFER WITH DATA OTHER THAN WHAT IS EXPECTED

6779 FILL READ DATA COMMAND

6797 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE
 INTO A REGISTER WHILE THE READ IS GOING ON

6831 NOW RMR IS SET BUT THE COMPLETION OF READ MUST BE
 WAITED ON

6842 CHANGE SAVED REGISTERS TO EXPECTED VALUE

6893 COMPARE REGISTERS BEFORE READ DATA WITH REGISTERS
 AFTER READ AND ATTEMPTED MODIFICATION OF REGISTER

6917 COMPARE DATA READ

6937 TEST 30 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2), 'RMR'

6939 A WRITE HEADER AND DATA COMMAND WILL BE GIVEN TO
 CYLINDER 1 SECTOR 0 TRACK 0 DATA WORDS
 OF 070707

A WRITE DATA COMMAND WILL BE GIVEN TO CYLINDER 1

6945 SECTOR 0, TRACK 0, 256 WORDS OF 2000
 AND 4 WORDS OF 2001. IMMEDIATELY AFTER GO
 AN ATTEMPT WILL BE MADE TO MODIFY A REGISTER
 RMR BIT #2 IN RHER1 SHOULD SET

AFTER THE WRITE IS COMPLETE ALL REGISTERS WILL
 BE CHECKED .

THE DATA WRITTEN WILL BE READ BACK AND CHECKED

THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST,
 RHER2, RHOF, RHCA, RHER3

6971 PREPARE TO WRITE HEADER AND DATA
6981 FILL WRITE FROM BUFFER WITH HEADER
6991 FILL WRITE FROM BUFFER WITH DATA
6999 FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
7009 FILL WRITE FROM BUFFER WITH WITH NEXT SECTOR DATA
7017 NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7056 ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
7065 NOW PREPARE FOR THE WRITE DATA COMMAND
7067 FILL WRITE FROM BUFFER WITH 256 OF 2000 AND 4 OF 2001
7081 FILL WRITE DATA COMMAND
7099 SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED
REGISTER MODIFICATION DURING A WRITE DATA
7133 NOW RMR MUST BE SET BUT THE COMPLETION OF
WRITE DATA MUST BE WAITED ON
7144 CHANGE SAVED REGISTERS TO EXPECTED VALUE
7195 COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
AFTER WRITE AND ATTEMPTED MODIFICATION OF REGISTER
7219 CLEAR ALL ERROR FLAGS
7227 FILL WRITE FROM BUFFER WITH EXPECTED DATA
7241 NOW THE READ DATA COMMAND WILL BE FILLED
7259 NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
7297 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
RD AND RS IMMEDIATELY AFTER GO IS ISSUED
7321 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7339 NOW COMPARE REGISTERS BEFORE READ DATA WITH
AFTER COMMAND
7355 NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
WAS GOOD

7370 IF ALL 7 REGISTERS NOT COMPLETE THEN REPEAT

7378 TEST 31 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2)RMR

7380 A READ DATA COMMAND IS GIVEN TO CYLINDER 0, SECTOR 0
TRACK 0. IMMEDIATELY AFTER GO RHAS IS WRITTEN INTO
WITH ALL ONES RMR BIT #2 IN RHER SHOULD NOT SET

7394 FILL WRITE FROM BUFFER WITH EXPECTED DATA

7401 FILL READ INTM BUFFER WITH ALL ONES

7408 NOW THE READ DATA COMMAND WILL BE FILLED

7425 NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

7459 TIME IS NOT IMPORTANT

7468 COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED

7492 NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

7514 NOW COMPARE REGISTERS BEFORE READ DATA WITH
AFTER COMMAND

7530 NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
WAS GOOD

7548 TEST 32 ILLEGAL FUNCTION RHER1 - (BIT #0,ILF)

7550 THIS WILL CALCULATE EVERY ILLEGAL FUNCTION
BETWEEN 0 AND 77. EACH TIME AN ILLEGAL FUNCTION
IS FORMED IT WILL BE STORED IN ILLEGAL THEN
EXECUTION OF ILLEGAL
WILL BE ATTEMPTED AND RESULTS CHECKED

7571 GENERATE ILLEGAL FUNCTION

7589 ILLEGAL FUNCTION HAS BEEN FOUND
IT IS IN 'ILLEGL'

7593 SAVE REGISTERS FOR COMPARISON AFTER GO

7662 CHANGE SAVED REGISTERS TO EXPECTED VALUE

7684 RHCS1 WILL HAVE SC AND TRE ADDED IF FUNCTION IS GREATER THAN 50

7695 RHCS2 WILL HAVE NOTHING ADDED IF FUNCTION IS LESS THAN 50

7719 NOW COMPARE REGISTERS AFTER GIVING AN ILLEGAL COMMAND

7747 TEST 33 OPERATION INCOMPLETE - RHER1(BIT #13)OPI
A WRITE HEADER AND DATA COMMAND IS GIVEN
CYLINDER 0 SECTOR 1 TRACK 0 KEYS 0 DATA 177777
WORDCOUNT 260

AFTER GO IS GIVEN THEN THREE INDEX PULSES ARE
GIVEN. THIS SHOULD BRING OPI HIGH

7764 THESE ARE REGULAR SETUPS

7776 SETUP DATA, WRITE HEADER & DATA, AND FORMAT OF THE WRITE

7791 SAVE REGISTERS FOR COMPARISON AFTER READ

7799 GO TO WRITE HEADER AND DATA
BUT BEFORE GO, ONE INDEX PULSE IS GIVEN
TO CLEAR OUT THE SECTOR CLOCK COUNTER IN THE RH11
SO THAT NO SECTOR PULSES COME DURING THIS TEST

7817 ISSUE THE FIRST DIAGNOSTIC INDEX PULSE

7821 SECOND INDEX PULSE

7825 THIRD INDEX PULSE

7830 CHANGE SAVED REGISTERS TO EXPECTED VALUE

7874 RHWC, RHBA AND OR AND IR BITS OF RHCS2 WILL NOT BE CHECKED

7883 COMPARE REGISTERS BEFORE WRITE WITH RESULTS AFTER WRITE

7904 TEST 34 CONSECUTIVE SECTOR FORMATTING

7906 46 CONSECUTIVE SECTORS WILL BE FORMATTED
STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
FORMATTING WILL BE DONE BY A WRITE HEADER AND
DATA COMMAND FOR 4 WORDS, ONE SECTOR
AT A TIME.

AFTER EACH SECTOR IS WRITTEN, 'SC' WILL BE CHECKED
TO INSURE THAT THERE WERE NO ERRORS

7928 FILL WRITE FROM BUFFER WITH THE HEADER

7939 NOW THE WRITE HEADER AND DATA COMMAND WILL BE SETUP

7981 ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC

7990 NOW ONE MORE SECTOR HAS BEEN WRITTEN
'SC' WILL BE CHECKED TO MAKE SURE
NO ERRORS OCCURED

8001 A SECTOR HAS BEEN FORMATTED NOW,
THE HARDWARE WILL BE CLEARED AND
CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.

8031 TEST 35 OPERATION INCOMPLETE - RHER1 (BIT #13)OPI

8033 THIS WILL TEST THAT OPI DOES NOT SET WHEN THREE NORMAL
INDEX PULSES ARE ENCOUNTERED IN A READ COMMAND

8036 FIRST 46 CONSECUTIVE SECTORS WILL BE FORMATTED
STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
FORMATTING WILL BE DONE BY A WRITE HEADER AND
DATA COMMAND FOR 4 WORDS, ONE SECTOR
AT A TIME

THEN A READ HEADER AND DATA WILL BE DONE
FOR CYLINDER 0 TRACK 0 SECTOR 21 FOR
11960 WORDS (260.X22X2+260+260) WITH BUS
ADDRESS INHIBIT SET.

AT THE END ALL REGISTERS WILL BE CHECKED.

8062 FILL WRITE FROM BUFFER WITH HEADER

8073 NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

8115 ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC

8124 NOW ONE MORE SECTOR HAS BEEN WRITTEN
'SC' WILL BE CHECKED TO MAKE SURE
NO ERRORS OCCURED

8138 ONE SECTOR HAS BEEN FORMATTED NOW,
THE HARDWARE WILL BE CLEARED AND
CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.

8165 NOW 46 SECTORS HAVE BEEN FORMATTED

8167 READ HEADER AND DATA FOR 46 SECTORS=11960 WORDS
WITH BUS ADDRESS INHIBITED

8176 FILL READ HEADER AND DATA COMMAND

8215 TIME IS NOT IMPORTANT

8225 NOW THAT ALL 11950 WORDS HAVE BEEN READ
'OPI' WILL BE CHECKED TO BE NOT SET

8236 'SC' WILL BE CHECKED

8250 TEST 36 HEAD SELECTION TEST ERR & TRE
THIS TESTS HEAD SELECTION LOGIC ONLY. A WRITE HEADER AND
DATA COMMAND IS GIVEN TO EACH TRACK FROM 0 TO 18 ON
CYLINDER 0, SECTOR 0.
THE DATA ON EACH SECTOR IS UNIQUE. THE LEAST SIGNIFICANT
5 BITS GIVE SECTOR THE NEXT LEAST SIGNIFICANT 5 BITS
GIVE TRACK THE NEXT 6 BITS GIVE CYLINDER

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

BETWEEN THE WRITE AND READ ONLY "ERR" AND "TRE" ARE CHECKED

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

8277 THE FOLLOWING CLEARS ARE TO INITIALIZE TEST FROM CYLINDER 0

8284 THIS GETS THE HEADS TO CYLINDER 0

8321 FILL WRITE FROM BUFFER WITH HEADER

8332 FILL WRITE FROM BUFFER WITH DATA

8341 THE WRITE HEADER AND DATA COMMAND WILL BE FILLED

8381 ONE REVOLUTION = 16670 MICRO SEC., ONE SECTOR = 760
MICRO SEC. MAX TIME ALLOWED = ONE REVOLUTION + HEAD
SWITCH + 2 SECTORS, MIN TIME ALLOWED = SECTOR (FIRST CASE)
IF THERE IS A FAILURE HERE HALT PROGRAM AFTER ERROR WITH
SWITCH IS AND SEE CURRENT CYLINDER REGISTER TO DETERMINE
WHAT CYLINDER IS FAILING

8396 NOW SECTOR 0 OF ONE TRACK HAS BEEN WRITTEN CHECK COMPOSIT
ERROR BIT TO BE SURE NO ERRORS HAPPENED

8399 SAVE REGISTERS IN SAVE TABLE

8411 THE FOLLOWING 3 ADDS SETS UP FOR NEXT TRACK WRITING

8420 THE FOLLOWING CLEARS SETS UP FOR READ HEADER AND DATA

8431 SET UP FOR READ HEADER AND DATA

8434 FILL READ INTO BUFFER WITH ALL ONES

8444 FILL WRITE FROM BUFFER WITH EXPECTED HEADER

8455 FILL WRITE FROM BUFFER WITH EXPECTED DATA

8464 FILL COMMAND FOR READ HEADER AND DATA

8512 NOW SECTOR 0 OF ONE TRACK HAS BEEN READ CHECK COMPOSIT
ERROR BIT TO BE SURE NO ERROR HAPPENED

8515 SAVE REGISTERS IN SAVE TABLE

8527 NOW THE READ DATA WILL BE COMPARED DATA IN EACH SECTOR
IS UNIQUE IF PROGRAM IS HALTED ON ERROR THEN LOOK AT
RHDST TO GET WHAT TRACK IS IN ERROR. LOOKING AT THE DATA
BITS NO 4 5 6 7 9 IN GOOD DATA ALSO GIVES TRACK NUMBER
IN GOOD DATA ALSO GIVES TRACK NUMBER

8554 NOW INCREMENT TO READ NEXT TRACK

8568 TEST 37 DIFFERENCE LINES

8570 A WRITE HEADER AND DATA WILL BE DONE ON ALL CYLINDERS
UP TO 256. 512. ON SECTOR ZERO, TRACK ZERO. THE DATA WILL
BE THE CYLINDER NUMBER

THEN A RECALIBRATE AND READ HEADER AND DATA WILL BE DONE
ON CYLINDERS 0,1,2,4,8,16,32,64,128,256 & 512.

DATA WILL BE CHECKED.
ON AN ERROR, LOOPING WILL BE DONE ON THE
READ ONLY

8596 SET UP TO INITIALIZE TEST FROM CYLINDER 0, TRACK 0,
SECTOR 0

8602 THIS IS TO GET THE HEADS TO CYLINDER ZERO

8626 THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE MAX.
CYLINDER DIFFERENCE IS SET UP

8647 FILL WRITE FROM BUFFER WITH HEADER

8658 FILL WRITE FROM BUFFER WITH DATA

8667 THE WRITE HEADER AND DATA COMMAND WILL BE LOADED

8707 ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760
MICRO SECONDS, ONE SEEK = 7000 MICRO SECONDS.
MAX TIME = 1 REVOLUTION + 1 SEEK + 2 SECTORS
MIN TIME = 1 SECTOR

8720 NOW ONE SECTOR WRITE IS COMPLETE. CHANGES WILL BE MADE
FOR THE NEXT SECTOR, THEN THE ABOVE WILL BE REPEATED
UNTIL CYLINDER 256. 512. IS REACHED

8729 NOW ALL 256. 512. CYLINDERS HAVE CYLINDER NUMBER WRITTEN
AS DATA ON SECTOR 0, TRACK 0. NOW A RECALIBRATE, FOLLOWED
BY READ HEADER AND DATA, THEN A CHECK WILL BE DONE ON
CYLINDER 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, AND 0

8737 INITIALIZE, RECALIBRATE, AND READ CYLINDERS

8740 SETUP FOR CYLINDER 0

8774 CLEAR READ INTO BUFFER WITH ALL ONES

8792 FILL WRITE FROM BUFFER WITH EXPECTED HEADER

8900 FILL READ HEADER AND DATA COMMAND

8832 ONE SECTOR = 760 MICRO SECONDS, ONE REVOLUTION =
16670 MICRO SECONDS, MAX SEEK = 5000 MICRO SECONDS
MAX TIME = ONE REV + 1 SEEK + 1 SECTOR
MIN TIME = 1 SECTOR

8853 CHECK READ WORDS AS ALL READ COMMANDS HAVE BEEN CHECKED

8855 (DATA ERRORS MAY IMPLY "IMPLIED SEEK" ERRORS)

8870 NOW ONE CYLINDER HAS BEEN CHECKED. CHANGES WILL BE MADE
TO READ THE NEXT CYLINDER AND THE ABOVE SECTOR READ WILL BE
REPEATED

8922 TEST 40 END OF DRIVE

8924 THIS IS THE END OF TEST FOR ONE DRIVE
IF THERE ARE MORE DRIVES, THEN THE PROGRAM
JUMPS TO TEST 5 FOR NEXT DRIVE TEST

8928 END OF PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE

8971

8972

SUBROUTINES

8973

8978

END OF PASS ROUTINE

8981 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

9037 NOW FILL DATA

9192 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

9278 THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS

9301 NOW TIME AND TOLERANCE WILL BE CHECKED

9363 THIS HAS THE TWO COUNT DOWNS FROM !7777

9389 BIT DID SET SO CHECK IF INTERRUPT OCCURRED

9391 THE AMOUNT OF TIME ALLOWED CAN BE CHANGED BY ALTERING LOCATION
"RPTCTR" ABOVE

9462 NOW DATA WILL BE FILLED

9501 NOW SAVE REGISTERS

9505 NOW COMPARES WILL MADE

9601 THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
 IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
 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

9703 THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
 ADDRESS FROM 176700 TO ANY TYPED VALUE

9775 *****
 SYSMAC LIBRARY ROUTINES

9778 *****
 SCOPE HANDLER ROUTINE

9781 THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 AND LOAD THE TEST NUMBER (STSTNM) INTO THE DISPLAY REG. (DISPLAY<7:0>)
 AND LOAD THE ERROR FLAG (SERFLG) 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
 SWCB=1 LOOP ON TEST IN SWR<7:0>
 CALL SCOPE ;:SCOPE=ICT

9841 *****
 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

9844 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: MC: NUM,-.SP) ;:PUT THE BINARY NUMBER ON THE STACK
 TYPDS ;:GO TO THE ROUTINE

9908

 TYPE ROUTINE

9911 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

9978

 TTY INPUT ROUTINE

9989 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

10006 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)

10058 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.

10073 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.

10144 THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 CALL:

ROCHR ;;GET A CHARACTER FROM THE QUEUE
 RETURN HERE ;;CHARACTER IS ON THE STACK
 ;;WITH PARITY BIT STRIPPED OFF

10169 THIS ROUTINE WILL INPUT A STRING FROM THE TTY
CALL:

RDLIN
RETURN HERE

::INPUT A STRING FROM THE TTY
::ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
::TERMINATOR WILL BE A BYTE OF ALL 0'S

10206

READ AN OCTAL NUMBER FROM THE TTY

10209 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
RETURN HERE

::READ AN OCTAL NUMBER
::LOW ORDER BITS ARE ON TOP OF THE STACK
::HIGH ORDER BITS ARE IN \$HIOCT

10259

ERROR HANDLER ROUTINE

10262 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

10310

ERROR MESSAGE TYPEOUT ROUTINE

10312 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.

[Handwritten marks and scribbles on the right margin]

10541

BINARY TO OCTAL (ASCII) AND TYPE

10544 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
```

10518

TRAP DECODER

10621 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.

10534

TRAP TABLE

10636 THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
BY THE "TRAP" INSTRUCTION.

10655

POWER DOWN AND UP ROUTINES

10702

46 61 172 182 193 210 247 305 1239 1412 1413 1414 1417 1432 151

*** ERROR TABLE, BIT DEFINITIONS & STARTING ADDRESSES ***

OPERATIONAL SWITCH SETTINGS
BASIC DEFINITIONS
TRAP CATCHER
ACT11 HOOKS
STARTING ADDRESS
MEMORY MANAGEMENT DEFINITIONS
COMMON TAGS
ERROR POINTER TABLE
REGISTER ADDRESSES

*** DIAGNOSTIC CODE ***

SETUP TESTS
INITIALIZE THE COMMON TAGS
GET VALUE FOR SOFTWARE SWITCH REGISTER
T1 REFERENCE EACH REGISTER
T2 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
T3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
T4 TYPE SERIAL NUMBER AND DRIVE TYPE
T5 CHECK MOL TO BE HIGH
T6 PROGRAM INTERRUPT
T7 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
T10 SET VV BIT #6 IN RHDS1
DATA TRANSFER RELATED ERRORS (USING MEDIA)
T11 LAST BLOCK TRANSFERED-RHDS1 LBT
T12 SEARCH COMMAND
T13 SEARCH COMMAND
T14 HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)
T15 HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)
T16 HEADER COMPARE ERROR - RHER1 HCE
T17 RESTORE SECTOR 1 CYLINDER 1 TRACK 1
T20 INVALID ADDRESS ERROR - RHER1 - 'IAE'
T21 INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE
T22 INVALID ADDRESS ERROR RHER1 -BIT #10
T23 INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE
T24 ADDRESS OVERFLOW ERROR - RHER1 (BIT#9) AOE
T25 FORMAT ERROR - RHER1 (BIT #4)FMT
T26 FORMAT ERROR - RHER1 (BIT #4)FMT
T27 REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR
T30 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2), 'RMR'
T31 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2)RMR
T32 ILLEGAL FUNCTION RHER1 - (BIT #0,ILF)
T33 OPERATION INCOMPLETE - RHER1(BIT #13)OPI
T34 CONSECUTIVE SECTOR FORMATTING
T35 OPERATION INCOMPLETE - RHER1 (BIT #13)OPI
T36 HEAD SELECTION TEST ERR & TRE
T37 DIFFERENCE LINES
T40 END OF DRIVE

SUBROUTINES

4-1
4-10
4-11
4-12
4-13
4-14
4-15
4-16
4-17
4-18
4-19
4-20
4-21
4-22
4-23
4-24
4-25
4-26
4-27
4-28
4-29
4-30
4-31
4-32
4-33
4-34
4-35
4-36
4-37
4-38
4-39
4-40
4-41
4-42
4-43
4-44
4-45
4-46
4-47
4-48
4-49
4-50
4-51
4-52
4-53
4-54
4-55
4-56
4-57
4-58
4-59
4-60
4-61
4-62
4-63
4-64
4-65
4-66
4-67
4-68
4-69
4-70
4-71
4-72
4-73
4-74
4-75
4-76
4-77
4-78
4-79
4-80
4-81
4-82
4-83
4-84
4-85
4-86
4-87
4-88
4-89
4-90
4-91
4-92
4-93
4-94
4-95
4-96
4-97
4-98
4-99
4-100

K04

MNDEC-11-DZRJJ-A, RPO4 5/6 FUNCT. CONT. TST-PT 2
DZRJJ.A.P11

MACY11 27(655) 30-MAR-76 18:59

SEQ 0048

TABLE OF CONTENTS

9214	END OF PASS ROUTINE
9611	SYSMAC LIBRARY ROUTINES
9914	SCOPE HANDLER ROUTINE
9978	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9945	TYPE ROUTINE
10015	TTY INPUT ROUTINE
10244	READ AN OCTAL NUMBER FROM THE TTY
10297	ERROR HANDLER ROUTINE
10318	ERROR MESSAGE TIMEOUT ROUTINE
10519	BINARY TO OCTAL (ASCII) AND TYPE
10656	TRAP DECODER
10672	TRAP TABLE
10693	POWER DOWN AND UP ROUTINES

M04

MNDEC-11-DZRJJ-A RFD4 5/6 FUNCT. CONT. TST-PT 2
DZRJJA.P11 OPERATIONAL SWITCH SETTINGS

MACY11 27(655) 30-MAR-76 18:59 PAGE 2

SEQ 0050

5500000000

10

BELL ON ERROR
LOOP ON ERROR
LOOP ON TEST IN SWR<7:0>
STOP FURTHER COMPARES IF SW08 IS LOW
TYPE ALL REG. WITH ERROR IF SW8 LOW

.SBTTL BASIC DEFINITIONS

001000

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

STACK= 1000

.EQUIV EMT,ERROR

::BASIC DEFINITION OF ERROR CALL

.EQLIV 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

177774

.EQUIV PS,PSW

177772

STKLMT= 177774

::STACK LIMIT REGISTER

177570

PIRG= 177772

::PROGRAM INTERRUPT REQUEST REGISTER

177570

DSWR= 177570

::HARDWARE SWITCH REGISTER

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

0000010
0000011
0000012
0000013
0000014

SW03 = 10
SW02 = 11
SW01 = 12
SW00 = 13
.EQUIV SW09 SW09
.EQUIV SW08 SW08
.EQUIV SW07 SW07
.EQUIV SW06 SW06
.EQUIV SW05 SW05
.EQUIV SW04 SW04
.EQUIV SW03 SW03
.EQUIV SW02 SW02
.EQUIV SW01 SW01
.EQUIV SW00 SW00

0000015
0000016
0000017
0000018
0000019
0000020
0000021
0000022
0000023
0000024
0000025
0000026
0000027
0000028
0000029
0000030
0000031
0000032
0000033
0000034
0000035
0000036
0000037
0000038
0000039
0000040
0000041
0000042
0000043
0000044
0000045
0000046
0000047
0000048
0000049
0000050
0000051
0000052
0000053
0000054
0000055
0000056
0000057
0000058
0000059
0000060
0000061
0000062
0000063
0000064
0000065
0000066
0000067
0000068
0000069
0000070
0000071
0000072
0000073
0000074
0000075
0000076
0000077
0000078
0000079
0000080
0000081
0000082
0000083
0000084
0000085
0000086
0000087
0000088
0000089
0000090
0000091
0000092
0000093
0000094
0000095
0000096
0000097
0000098
0000099
0000100

BIT DEFINITIONS (BIT00 TO BIT15)

BIT15 = 100000
BIT14 = 100001
BIT13 = 100002
BIT12 = 100003
BIT11 = 100004
BIT10 = 100005
BIT09 = 100006
BIT08 = 100007
BIT07 = 100008
BIT06 = 100009
BIT05 = 100010
BIT04 = 100011
BIT03 = 100012
BIT02 = 100013
BIT01 = 100014
BIT00 = 100015
.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

ERRVEC = 4 : TIME OUT AND OTHER ERRORS
RESVEC = 10 : RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC = 14 : "T" BIT
TRIVEC = 14 : TRACE TRAP
BPTVEC = 14 : BREAKPOINT TRAP (BPT)
IOTVEC = 20 : INPUT/OUTPUT TRAP (IOT) **SCOPE**
PHRVEC = 24 : POWER FAIL
EMTVEC = 30 : EMULATOR TRAP (EMT) **ERROR**
TRAPVEC = 34 : "TRAP" TRAP
TKVEC = 60 : TTY KEYBOARD VECTOR

169
170
171

0C0064
000240

TPVEC= 64
PIRQVEC=240

::TTY PRINTER VECTOR
::PROGRAM INTERRUPT REQUEST VECTOR

.SBTTL TRAP CATCHER

000000

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

000174 000174
000175 000000

. = 174
DISPREG: .WORD 0 ;:SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;:SOFTWARE SWITCH REGISTER

.SBTTL ACT11 HOOKS

::*****
:HOOKS REQUIRED BY ACT11

000200
000046
000046 000046
000052 000052
000052 020000
000200

SSVPC= ;SAVE PC
. = 46 ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SECP
SENDAD
. = 52 ;:2)SET LOC.52 TO 20000
.WORD 20000 ;:RESTORE PC
.=SSVPC

.SBTTL STARTING ADDRESS

000200 000200
000200 000137 004710
000210 000210
000210 000137 004674
000220 000220
000220 000137 004660

. = 200
JMP @BEGIN ;NORMAL START
. = 210
JMP @BEGIN2 ;JUMP TO SELECT DRIVE START
. = 220
JMP @BEGIN1 ;JUMP TO NO OPERATOR TESTS START

::*STARTING ADDRESS 200 FOR NORMAL STARTS
:*THIS WILL TEST ALL RPC4'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
:*
:*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
:*
:*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
:*AT THE DRIVE

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099

.SBTTL MEMORY MANAGEMENT DEFINITIONS

;*KT11 VECTOR ADDRESS

000250

MMVEC= 250

;*KT11 STATUS REGISTER ADDRESSES

177572
177574
177576
172516

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

;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

172300
172302
172304
172306
172310
172312
172314
172316

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

;*KERNEL "I" PAGE ADDRESS REGISTERS

172340
172342
172344
172346
172350
172352
172354
172356

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

001110

.=1110

: ' 2

R-111

G05

MNDEC-1:-DZRJJ-A, RPO4.5.6 FUNCT. CONT. TST-PT 2
DZRJJ.A.P11 COMMON TAGS

MACY11 27(655) 30-MAR-76 18:59 PAGE 9

SEQ 0057

301 001222 077
302 001223 015
303 001224 000012
304

\$QUES: .ASCII <2>
\$CRLF: .ASCII <15>
\$LF: .ASCII <12>
::QUESTION MARK
::CARRIAGE RETURN
::LINE FEED
::*****

.SBTTL ERROR POINTER TABLE

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

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

001226

\$ERRTB:

:ITEM1

001226 042244

EM1

:RPO4 DID NOT INTERRUPT
:WAITED ON BIT DID NOT OCCUR
:PC

001230 057134

DH1

:WAT PC
:BIT WAITED
:REG ADDRESS
:REG CONTENTS
:RHCSI CONTENTS
:\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CSI
:0,0,0,0,0,0

001232 061404

DT1

001234 061724

DF1

:ITEM2

001236 042113

EM2

: INTERRUPT ENABLE BIT DOWN BUT
: WAITED ON BIT DID NOT OCCUR
: PC

001240 057134

DH1

: WAT PC
: BIT WAITED
: REG ADDRESS
: REG CONTENTS
: RHCSI CONTENTS
: \$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CSI
: 0,0,0,0,0,0

001242 061404

DT1

001244 061724

DF1

:ITEM3

001246 042362

EM3

: RPO4 DID NOT INTERRUPT WHEN
: WAITED ON BIT DID SET
: PC

001250 057134

DH1

: WAT PC
: BIT WAITED
: REG ADDRESS
: REG CONTENTS
: RHCSI CONTENTS
: \$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, CSI
: 0,0,0,0,0,0

001252 061404

DT1

001254 061724

DF1

:ITEM4

305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

359	001256	042443	EM4	:WAITED ON BIT DID SET BUT
360				:TIME IS IN ERROR
361				:TIME IS GIVEN IN 10 MICRO SEC.
362				: (DECIMAL)
363	001260	057314	DH4	:PC
364				:WAT PC
365				:BIT WAITED
366				:REG ADDRESS
367				:TIME IN 10 MSEC
368	001262	061424	DT4	:SERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, WAITIM
369	001264	061733	DF4	:0,0,0,0,0,1
370				
371				:ITEM5
372	001266	042554	EM5	:RHAS DOES NOT CLEAR BY
373				:MOVING IN ALL ONES
374	001270	057455	DH5	:PC
375				:REG. ADDR.
376				:GOOD DATA
377				:RECEIVED DATA
378	001272	061446	DT5	:SERRPC, REGADR, \$GDDAT, \$BDDAT
379	001274	061742	DF5	:0,0,0,0
380				
381				:ITEM6
382	001276	042626	EM6	:LOADING RHER1 FOR ALL
383				:UNITS DID NOT SET ANY BITS
384				:IN RHAS-NO UNITS PRESENT
385	001300	057574	DH6	:PC
386				:REG ADDR
387				:RECEIVED DATA
388	001302	061462	DT6	:SERRPC, REGADR, \$BDDAT
389	001304	061747	DF6	:0,0,0
390				
391				:ITEM7
392	001306	042714	EM7	:SPECIFIED REGISTER NONEXISTANT
393				:SO ABORT PROGRAM
394	001310	057673	DH7	:PC
395				:ADDR. OF REG.
396	001312	061474	DT7	:SERRPC, TEMPI
397	001314	061753	DF7	:0,0
398				
399				:ITEM10
400	001316	042776	EM10	:STOPED DRIVE HAS MOL BIT
401				:IN RHDS1 = 1
402	001320	057733	DH10	:PC
403				:TEST NO
404				:FAILING REG ADDR
405				:CONTENTS OF RHCS1
406				:CONTENTS OF RHCS2
407				:CONTENTS OF RHDS1
408				:CONTENTS OF RHER1
409	001322	061504	DT10	:SERRPC, \$STNM, \$BDADR, \$S1, \$S2, \$S1, \$E1
410	001324	061756	DF10	:0,0,0,0,0,0,0
411				
412				:ITEM11

413	001326	043045	EM11	:WITH SPINDLE POWERED DOWN
414				:RHCS2 SHOULD HAVE ONLY
415				:UNIT NUMBER AND IR HIGH
416	001330	057733	DH10	:PC
417				:TEST NO
418				:FAILING REG. ADR
419				:CONTENTS OF RHCS1
420				:CONTENTS OF RHCS2
421				:CONTENTS OF RHDS1
422				:CONTENTS OF RHER1
423	001332	061504	DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
424	001334	061756	DF10	:0,0,0,0,0,0,0
425				
426				:ITEM12
427	001336	043152	EM12	:AFTER A POWER UP WITH
428				:NO PACK ACKNOWLEDGE COMMAND
429				:RHDS1 SHOULD HAVE MOL=1, VV=0
430	001340	057733	DH10	:PC
431				:TEST NO
432				:FAILING REGISTER ADDR.
433				:CONTENTS OF RHCS1
434				:CONTENTS OF RHCS2
435				:CONTENTS OF RHDS1
436				:CONTENTS OF RHER1
437	001342	061504	DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
438	001344	061756	DF10	:0,0,0,0,0,0,0
439				
440				:ITEM13
441	001346	043260	EM13	:AFTER A POWER UP WITHOUT
442				:ANY INIT RHCS1 SHOULD
443				:HAVE GO=0, DVA=1, RDY=1
444				:IE=0, DISREGARD
445				:ALL OTHER BITS
446	001350	057733	DH10	:PC
447				:TEST NO
448				:FAILING REGISTER ADDR.
449				:CONTENTS OF RHCS1
450				:CONTENTS OF RHCS2
451				:CONTENTS OF RHDS1
452				:CONTENTS OF RHER1
453	001352	061504	DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
454	001354	061756	DF10	:0,0,0,0,0,0,0
455				
456				:ITEM14
457	001356	043377	EM14	:AFTER POWER UP RHCC
458				:SHOULD BE=0
459	001360	057455	DH5	:PC
460				:REG. ADDR.
461				:GOOD DATA
462				:RECEIVED DATA
463	001362	061446	DT5	:SERRPC,REGADR,\$GDCAT,\$BDCAT
464	001364	061742	DF5	:0,0,0,0
465				

466			:ITEM15		
467	001366	043451		EM15	:PACK ACKNOWLEDGE CAUSED
468					:AN ERROR
469					:GOOD DATA IS BEFORE COMMAND
470					:RECEIVED DATA IS AFTER COMMAND
471	001370	057455		DH5	:PC
472					:REG. ADDR.
473					:GOOD DATA
474					:RECEIVED DATA
475	001372	061446		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
476	001374	061742		DFS	:0,0,0,0
477					
478			:ITEM16		
479	001376	043612		EM16	:GIVING A NO-OP COMMAND CAUSED
480					:AN ERROR
481					:GOOD DATA GIVES REGISTER
482					:CONTENTS BEFORE COMMAND
483					:RECEIVED DATA GIVES REGISTER
484					:CONTENTS AFTER COMMAND
485	001400	057455		DH5	:PC
486					:REG. ADDR.
487					:GOOD DATA
488					:RECEIVED DATA
489	001402	061446		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
490	001404	061742		DFS	:0,0,0,0
491					
492			:ITEM17		
493	001406	043740		EM17	:DRIVE CLEAR COMMAND
494					:CAUSED AN ERROR
495					:GOOD DATA GIVES WHAT SHOULD
496					:BE THERE
497					:RECEIVED DATA GIVES WHAT WAS
498					:THERE AFTER COMMAND
499	001410	057455		DH5	:PC
500					:REG. ADDR.
501					:GOOD DATA
502					:RECEIVED DATA
503	001412	061446		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
504	001414	061742		DFS	:0,0,0,0
505					
506			:ITEM20		
507	001416	044075		EM20	:READ-IN COMMAND GAVE AN ERROR
508					:GOOD DATA HAS WHAT SHOULD BE THERE
509					:RECEIVED DATA HAS WHAT WAS
510					:AFTER COMMAND
511	001420	057455		DH5	:PC
512					:REG. ADDR.
513					:GOOD DATA
514					:RECEIVED DATA
515	001422	061446		DT5	:\$ERRPC,REGADR,\$GDDAT,\$BDDAT
516	001424	061742		DFS	:0,0,0,0
517					
518			:ITEM 21		
519					

520	001426	044244	EM21	:RHCS1 CONTENTS DURING :COMMAND WAS IN ERROR
521				
522	001430	057455	DH5	
523	001432	061446	DT5	
524	001434	061742	DF5	
525				
526				
527				
528	001436	044317	:ITEM 22 EM22	:RHDS1 CONTENTS DURING :COMM ANS WAS IN ERROR
529	001440	057455	DH5	
530	001442	061446	DT5	
531	001444	061742	DF5	
532				
533				
534	001446	044372	:ITEM 23 EM23	:UNLOAD COMMAND GAVE AN ERROR :GOOD DATA GIVES WHAT SHOULD :BE THERE :RECEIVED DATA GIVES WHAT WAS :THERE AFTER COMMAND
535				
536				
537				
538				
539	001450	057455	DH5	
540	001452	061446	DT5	
541	001454	061742	DF5	
542				
543				
544	001456	044541	:ITEM 24 EM24	:OFFSET COMMAND CAUSED AN ERROR :GOOD DATA IS WHAT SHOULD BE THERE :RECEIVED DATA GIVES WHAT WAS THERE :AFTER AN OFFSET COMMAND
545				
546				
547				
548	001460	057455	DH5	
549	001462	061446	DT5	
550	001464	061742	DF5	
551				
552				
553	001466	044704	:ITEM 25 EM25	:RETURN TO CENTER LINE COMMAND :CAUSED AN ERROR :GOOD DATA GIVES WHAT SHOULD BE :THERE :RECEIVED DATA GIVES WHAT WAS :THERE AFTER COMMAND
554				
555				
556				
557				
558				
559	001470	057455	DH5	
560	001472	061446	DT5	
561	001474	061742	DF5	
562				
563				
564	001476	045066	:ITEM 26 EM26	:500 OFFSETS CAUSED AN ERROR
565	001500	060112	DH26	:PC
566				:CONT. OF RHCS1
567				:CONT. OF RHCS2
568				:CONT. OF RHDS1
569				:CONT. OF RHER1
570				:CONT. OF RHER2
571				:CONT. OF RHER3
572	001502	061524	DT26	:SERRPC,CS1,CS2,DS1,ER1,ER2,ER3
573	001504	061765	DF26	:0,0,0,0,0,0

22

574					
575			: ITEM 27		
576	001506	045156	EM27		: WRITE HEADER AND DATA
577					: CAUSED IMPROPER REGISTER CHANGE
578					: GOOD DATA GIVES WHAT
579					: SHOULD BE THERE
580					: RECEIVED DATA GIVES WHAT
581					: WAS THERE AFTER COMMAND
582	001510	057455	DH5		
583	001512	061446	DT5		
584	001514	061742	DF5		
585					
586			: ITEM 30		
587	001516	045374	EM30		: WRITE HEADER AND DATA
588					: CHANGED WRITE FROM BUFFER
589	001520	060312	DH30		: PC
590					: WORD NO
591					: GOOD DATA
592					: BAD DATA
593	001522	061546	DT30		: \$ERRPC, ERWORD, \$GDDAT, \$BDDAT
594	001524	061775	DF30		: 0,0,0.0
595					
596			: ITEM 31		
597	001526	045454	EM31		: READ HEADER AND DATA CAUSED
598					: IMPROPER REGISTER CHANGE
599					: GOOD DATA HAS WHAT SHOULD
600					: BE THERE
601					: RECEIVED DATA GIVES WHAT
602					: WAS THERE AFTER COMMAND
603	001530	057455	DH5		
604	001532	061446	DT5		
605	001534	061742	DF5		
606					
607			: ITEM 32		
608	001536	045671	EM32		: WRITE HEADER AND DATA FOLLOWED
609					: BY A READ HEADER AND DATA
610					: CAUSED A READ/WRITE ERROR
611	001540	060312	DH30		
612	001542	061546	DT30		
613	001544	061775	DF30		
614					
615			: ITEM 33		
616	001546	045776	EM33		: READ DATA CAUSED IMPROPER REGISTER
617					: CHANGE
618					: GOOD DATA GIVES WHAT SHOULD BE THERE
619					: RECEIVED DATA GIVES WHAT WAS THERE AFTER
620					: COMMAND
621	001550	057455	DH5		
622	001552	061446	DT5		
623	001554	061742	DF5		
624					
625			: ITEM 34		
626	001556	046200	EM34		: READ DATA INCORRECT
627	001560	060312	DH30		

628	001562	061546		DF30	
629	001564	061775		DF30	
630					
631			: ITEM 35		
632	001566	046224		EM35	: WRITE DATA COMMAND CAUSED
633					: IMPROPER REGISTER CHANGE
634					: GOOD DATA GIVES WHAT SHOULD BE THERE
635					: RECEIVED DATA GIVES REGISTER
636					: CONTENTS AFTER WRITE DATA
637	001570	057455		DH5	
638	001572	061446		DT5	
639	001574	061742		DF5	
640					
641			: ITEM 36		
642	001576	046442		EM36	: WRITE DATA COMMAND CHANGED
643					: WRITE FROM BUFFER
644	001600	060312		DH30	
645	001602	061546		DT30	
646	001604	061775		DF30	
647					
648			: ITEM 37		
649	001606	046517		EM37	: SEEK COMMAND CAUSED AN
650					: ERROR
651					: GOOD DATA GIVES WHAT SHOULD
652					: BE THERE
653					: RECEIVED DATA GIVES WHAT
654					: WAS THERE AFTER SEEK COMMAND
655	001610	057455		DH5	
656	001612	061446		DT5	
657	001614	061742		DF5	
658					
659			: ITEM 40		
660	001616	046734		EM40	: WRITE CHECK CAUSED AN
661					: IMPROPER REGISTER CHANGE
662					: GOOD DATA GIVES WHAT SHOULD
663					: BE THERE
664					: RECEIVED DATA GIVES WHAT WAS
665					: THERE AFTER COMMAND
666	001620	057455		DH5	
667	001622	061446		DT5	
668	001624	061742		DF5	
669					
670			: ITEM 41		
671	001626	047143		EM41	: LOCKING OUT WRITES BY WRITE
672					: LOCK BUTTON CAUSED IMPROPER
673					: REGISTER CHANGE
674					: GOOD DATA GIVES WHAT SHOULD
675					: BE THERE
676					: RECEIVED DATA GIVES WHAT
677					: WAS THERE AFTER WRITES
678					: WERE LOCKED OUT BY
679					: BUTTON
680	001630	057455		DH5	
681	001632	061446		DT5	

001634	061742	DFS	
		: ITEM 42	
001636	047424	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	061745	DFS	
001644	061745	DFS	
		: ITEM 43	
001646	047732	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	061730	DFS	
001654	061730	DFS	
		: ITEM 44	
001656	050240	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	061745	DFS	
001662	061745	DFS	
001664	061742	DFS	
		: ITEM 45	
001666	050532	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	061745	DFS	
001672	061746	DFS	
001674	061742	DFS	
		: ITEM 46	
001676	051040	EM46	: DATA READ FROM LAST

:WAS THERE AFTER ATTEMPTED
:WRITE DATA

001740 057455
001742 061446
001744 061742

DH5
DT5
DF5

:ITEM 53
EM53

:READ HEADER AND DATA AFTER
:A SEARCH CAUSED AN ERROR

001746 052403
001750 060312
001752 061546
001754 061775

DH30
DT30
DF30

:ITEM 54
EM54

:ATTEMPTED OPERATION WITH
:INVALID ADDRESS CAUSED
:IMPROPER REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD
:BE THERE
:RECEIVED DATA GIVES WHAT WAS
:THERE AFTER OPERATION

001760 057455
001762 061446
001764 061742

DH5
DT5
DF5

:ITEM 55
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

001766 052736

001770 057455
001772 061446
001774 061742

DH5
DT5
DF5

:ITEM 56
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

001776 053224

002000 060312
002002 061546
002004 061775

DH30
DT30
DF30

:ITEM 57
EM57

:ATTEMPTING DATA COMMAND
:WITH WRONG FORMAT BIT CAUSED
:IMPROPER REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD BE

002006 053434

000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097

002010 057455
002012 061446
002014 061742

DH5
DTS
DFS

: THERE
: RECEIVED DATA GIVES WHAT WAS
: THERE AFTER ATTEMPTED DATA
: TRANSFER

002016 053726

: ITEM 60
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 060563

DH60

: PC
: REG. ADDR.
: GOOD DATA
: RECEIVED DATA
: MODIFYING REGISTER
: \$ERRPC, REGADR, \$GDDAT, \$BDDAT, \$BDADR
: 0.0.0,0.0

002022 061600
002024 062010

DT60
DF60

: ITEM 61

002026 054335
002030 060720

EM61
DH61

: DEVICE NOT AVAILBE BEFOR COMMAND WAS TO BE GIVEN
: PC
: PC OF JSR
: RHDS1
: \$ERRPC, PCJSR, \$BDADR
: 0.0.0

002032 061616
002034 062016

DT61
DF61

: ITEM 62

002036 054335
002040 061012

EM62
DH62

: DEVICE NOT AVAILBE BEFOR COMMAND WAS TO BE GIVEN
: PC
: PC OF JSR
: RHDS1 WAS
: \$ERRPC, PCJSR, \$BDADR
: 0.0.0

002042 061630
002044 062022

DT62
DF62

: ITEM 63

002046 054421
002050 057455
002052 061446
002054 061742

EM63
DH5
DTS
DFS

: RHDS1 CONTENTS DURING
: COMMAND WAS IN ERROR

: ITEM 64

002056 054474

EM64

: RECALIBRATE COMMAND CAUSED
: IMPROPER REGISTER CHANGE.
: GOOD DATA GIVES WHAT SHOULD BE

: THERE.
: RECEIVED DATA GIVES WHAT WAS THERE
: AFTER COMMAND

002060 057455 D45
002062 061446 DT5
002064 061742 DF5

: ITEM65

002066 054713 EM65
002070 061065 DM65

: INTERRUPT FAILING
: PC
: TEST NO
: CONTENTS OF RHCS1
: CONTENTS OF RHAS
: CONTENTS OF RHDS1
: SERRPC, TSTNM, CS1, AS, DS1
: 0.0,0.0,0

002072 061642 D765
002074 062026 DF65

: ITEM66

002076 054735 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 061201 DM66

: PC
: RHDST
: RHER1
: RHER2
: RHER3
: RHCS1
: RHCS2
: SERRPC, DST, ER1, ER2, ER3, CS1, CS2
: 0.0,0.0,0.0,0

002102 061656 DT66
002104 062033 DF66

: ITEM67

002106 055146 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 060312 DH30
002112 061546 DT30
002114 061779 DF30

: ITEM70

002116 055422 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

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099

1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200

:RH11 REGISTER BITS

002166 000254

RPVEC: 254 ;RPO4 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)

000000
000002
000004
000006
000008
000010
000012
000014
000016
000018
000020
000022
000024
000026
000028
000030
000032
000034
000036
000038
000040
000042
000044
000046
000048
000050
000052
000054
000056
000058
000060
000062
000064
000066
000068
000070
000072
000074
000076
000078
000080
000082
000084
000086
000088
000090
000092
000094
000096
000098
000100
000102
000104
000106
000108
000110
000112
000114
000116
000118
000120
000122
000124
000126
000128
000130
000132
000134
000136
000138
000140
000142
000144
000146
000148
000150
000152
000154
000156
000158
000160
000162
000164
000166
000168
000170
000172
000174
000176
000178
000180
000182
000184
000186
000188
000190
000192
000194
000196
000198
000200
000202
000204
000206
000208
000210
000212
000214
000216
000218
000220
000222
000224
000226
000228
000230
000232
000234
000236
000238
000240
000242
000244
000246
000248
000250
000252
000254
000256
000258
000260
000262
000264
000266
000268
000270
000272
000274
000276
000278
000280
000282
000284
000286
000288
000290
000292
000294
000296
000298
000300
000302
000304
000306
000308
000310
000312
000314
000316
000318
000320
000322
000324
000326
000328
000330
000332
000334
000336
000338
000340
000342
000344
000346
000348
000350
000352
000354
000356
000358
000360
000362
000364
000366
000368
000370
000372
000374
000376
000378
000380
000382
000384
000386
000388
000390
000392
000394
000396
000398
000400
000402
000404
000406
000408
000410
000412
000414
000416
000418
000420
000422
000424
000426
000428
000430
000432
000434
000436
000438
000440
000442
000444
000446
000448
000450
000452
000454
000456
000458
000460
000462
000464
000466
000468
000470
000472
000474
000476
000478
000480
000482
000484
000486
000488
000490
000492
000494
000496
000498
000500
000502
000504
000506
000508
000510
000512
000514
000516
000518
000520
000522
000524
000526
000528
000530
000532
000534
000536
000538
000540
000542
000544
000546
000548
000550
000552
000554
000556
000558
000560
000562
000564
000566
000568
000570
000572
000574
000576
000578
000580
000582
000584
000586
000588
000590
000592
000594
000596
000598
000600
000602
000604
000606
000608
000610
000612
000614
000616
000618
000620
000622
000624
000626
000628
000630
000632
000634
000636
000638
000640
000642
000644
000646
000648
000650
000652
000654
000656
000658
000660
000662
000664
000666
000668
000670
000672
000674
000676
000678
000680
000682
000684
000686
000688
000690
000692
000694
000696
000698
000700
000702
000704
000706
000708
000710
000712
000714
000716
000718
000720
000722
000724
000726
000728
000730
000732
000734
000736
000738
000740
000742
000744
000746
000748
000750
000752
000754
000756
000758
000760
000762
000764
000766
000768
000770
000772
000774
000776
000778
000780
000782
000784
000786
000788
000790
000792
000794
000796
000798
000800
000802
000804
000806
000808
000810
000812
000814
000816
000818
000820
000822
000824
000826
000828
000830
000832
000834
000836
000838
000840
000842
000844
000846
000848
000850
000852
000854
000856
000858
000860
000862
000864
000866
000868
000870
000872
000874
000876
000878
000880
000882
000884
000886
000888
000890
000892
000894
000896
000898
000900
000902
000904
000906
000908
000910
000912
000914
000916
000918
000920
000922
000924
000926
000928
000930
000932
000934
000936
000938
000940
000942
000944
000946
000948
000950
000952
000954
000956
000958
000960
000962
000964
000966
000968
000970
000972
000974
000976
000978
000980
000982
000984
000986
000988
000990
000992
000994
000996
000998
001000

INT = 1
BUS = 2
MCH = 4
MTR = 8
MXP = 16
FCR = 32
NEM = 64
NED = 128
NCR = 256
OFT = 512

:UNIT SELECT (BIT #0)
:UNIT SELECT (BIT #1)
:UNIT SELECT (BIT #2)
:BUS ADDRESS INCREMENT INHIBIT (BIT #3)
:UNIBUS B DC LO (BIT #4)
:CLEAR (BIT #5)
:INPUT READY (BIT #6)
:OUTPUT READY (BIT #7)
:MASS BUS PARITY ERROR (BIT #8)
:MISSED TRANSFER ERROR (BIT #9)
:PROGRAM ERROR (BIT #10)
:NON EXISTANT MEMORY (BIT #11)
:NON EXISTANT DRIVE (BIT #12)
:UNIBUS PARITY ERROR (BIT #13)
:WRITE CHECK ERROR (BIT #14)
:DATA LATE (BIT #15)

:DATA BUFFER REGISTER (RHCB)
:EACH BIT IS CALLED BY BIT NUMBER

:RPO4 REGISTER BITS

:CONTROL AND STATUS REGISTER. (#00)

000000	GO =	1	:GO (BIT #0)
000001	IE =	100	:INTERRUPT ENABLE (BIT #6)
000002	RDY =	200	:READY (BIT #7)
000003	H16 =	400	:HIGH ORDER UNIBUS BITS (BIT #8)
000004	H17 =	1000	:HIGH ORDER UNIBUS BITS (BIT #9)
000005	PSEL =	2000	:PORT SELECT (BIT #10)
000006	DVA =	4000	:DEVICE AVAILABLE (BIT #11)
000007	MOPE =	20000	:MASSBUSS PARITY ERROR (BIT #13)
000008	TRE =	40000	:TRANSFER ERROR (BIT #14)
000009	SC =	100000	:SPECIAL CONDITION (BIT #15)

:STATUS REGISTER (RMS1) (#01)

000000	OFF5 =	1	:DRIVE FORWARD 5"/SEC. (BIT #0)
000001	OFF20 =	2	:DRIVE FORWARD 20"/SEC. (BIT #1)
000002	OFF64 =	4	:DRIVE TO INNER GAVRD BAND (BIT #2)
000003	GRV =	10	:GO REVERSE (BIT #3)
000004	CL64 =	20	:DIFFERENCE LESS THAN 64 (BIT #4)
000005	DE1 =	40	:DIFFERENCE EQUALS 1 (BIT #5)
000006	VV =	100	:VOLUME VALID (BIT #6)
000007	CRY =	200	:DRIVE READY (BIT #7)
000008	OPR =	400	:DRIVE PRESENT (BIT #8)
000009	PRG =	1000	:PROGRAMABLE (BIT #9)
000010	LBT =	2000	:LAST SECTOR TRANSFERRED (BIT #10)
000011	WFL =	4000	:WRITE LOCK (BIT #11)
000012	MOL =	10000	:MEDIUM ON-LINE (BIT #12)
000013	PIP =	20000	:POSITIONING OPERATION IN PROGRESS (BIT #13)
000014	ERR =	40000	:COMPOSIT ERROR. (BIT #14)
000015	ATA =	100000	:ATTENTION ACTIVE (BIT #15)

:ERROR REGISTER #01 (RHER1) (#02)

000000	ILF =	1	:ILLEGAL FUNCTION (BIT #0)
000001	ILR =	2	:ILLEGAL REGISTER (BIT #1)
000002	RMR =	4	:REGISTER MODIFICATION REFUSED (BIT #2)
000003	PAR =	10	:PARITY ERROR (BIT #3)
000004	FER =	20	:FORMAT ERROR (BIT #4)
000005	WCF =	40	:WRITE CLOCK FAIL (BIT #5)
000006	ECH =	100	:ECC HARD ERROR (BIT #6)
000007	HCE =	200	:HEADER COMPARE ERROR (BIT #7)
000008	HCR =	400	:HEADER CRC ERROR (BIT #8)
000009	ACE =	1000	:ADDRESS OVERFLOW ERROR (BIT #9)
000010	IAE =	2000	:INVALID ADDRESS ERROR (BIT #10)
000011	WLE =	4000	:WRITE LOCK ERROR (BIT #11)
000012	OTE =	10000	:DRIVE TIMING ERROR (BIT #12)
000013	OPT =	20000	:OPERATION INCOMPLETE (BIT #13)
000014	UNS =	40000	:DRIVE UNSAFE (BIT #14)

105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

1106      100000      DCK=      100000      ;DATA CHECK ERROR (BIT 15)
1107
1108      ;MAINTAINABILITY REGISTER (RHMR) (#03)
1109
1110      000001      DMD=      1      ;DIAGINOSTIC MODE (BIT #0)
1111      000002      MCLK=     2      ;MAINTAINABILITY CLOCK (BIT #1)
1112      000004      MINX=     4      ;MAINTAINABILITY INDEX (BIT #2)
1113      000010      MSTCK=    10     ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
1114      000020      MRD=     20     ;MAINTAINABILITY READ (BIT #4)
1115      000040      MWR=     40     ;MAINTAINABILITY WRITE (BIT #5)
1116      001000      DTSY=    1000    ;MAINTAINABILITY SYNC DETECTED (BIT #9)

```

```

1117      ;ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
1118
1119      000001      AT0=      1      ;DEVICE 0 (BIT #0)
1120      000002      AT1=      2      ;DEVICE 1 (BIT #1)
1121      000004      AT2=      4      ;DEVICE 2 (BIT #2)
1122      000010      AT3=     10     ;DEVICE 3 (BIT #3)
1123      000020      AT4=     20     ;DEVICE 4 (BIT #4)
1124      000040      AT5=     40     ;DEVICE 5 (BIT #5)
1125      000100      AT6=    100     ;DEVICE 6 (BIT #6)
1126      000200      AT7=    200     ;DEVICE 7 (BIT #7)

```

```

1127      ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
1128      ;EACH BIT IS CALLED BY BIT NUMBER

```

```

1129      ;DRIVE TYPE REGISTER (RHDT) (#06)
1130      ;EACH BIT IS CALLED BY BIT NUMBER

```

```

1131      ;LOOK-AHEAD REGISTER (RHLA) (#07)
1132
1133      000001      EXT1=      1      ;EXTENSION 1 (BIT #0)
1134      000002      EXT2=      2      ;EXTENSION 2 (BIT #1)
1135      000004      EXT4=      4      ;EXTENSION 3 (BIT #2)
1136      000010      EXT10=    10     ;EXTENSION 4 (BIT #3)
1137      000020      EXT20=    20     ;EXTENSION 5 (BIT #4)
1138      000040      EXT40=    40     ;EXTENSION 6 (BIT #5)
1139      000100      SC1=     100     ;SECTOR COUNT FIELD 0 (BIT #6)
1140      000200      SC2=     200     ;SECTOR COUNT FIELD 1 (BIT #7)
1141      000400      SC4=     400     ;SECTOR COUNT FIELD 2 (BIT #8)
1142      001000      SC10=    1000    ;SECTOR COUNT FIELD 3 (BIT #9)

```

1160	002000	SC20= 2000	;SECTOR COUNT FIELD 4 (BIT #10)
1161	004000	TRK1= 4000	;TRACK FIELD 1 (BIT #11)
1162	010000	TRK2= 10000	;TRACK FIELD 2 (BIT #12)
1163	020000	TRK4= 20000	;TRACK FIELD 3 (BIT #13)
1164	040000	TRK10= 40000	;TRACK FIELD 4 (BIT #14)
1165	100000	TRK20= 100000	;TRACK FIELD 5 (BIT #15)
1166			
1167			
1168			
1169			
1170	000001	WCU= 1	;WRITE CURRENT UNSAFE (BIT #0)
1171	000002	CSF= 2	;CURRENT SINK FAILURE (BIT #1)
1172	000004	WSU= 4	;WRITE SELECT UNSAFE (BIT #2)
1173	000010	CSU= 10	;CURRENT SWITCH UNSAFE (BIT #3)
1174	000020	MSE= 20	;MOTOR SEQUENCE ERROR (BIT #4)
1175	000040	TDF= 40	;TRANSITIONS DETECTOR FAILURE (BIT #5)
1176	000100	TUF= 100	;TRANSITIONS UNSAFE (BIT #6)
1177	000200	FEN= 200	;FAILSAFE ENABLED (BIT #7)
1178	000400	WRU= 400	;WRITE READY UNSAFE (BIT #8)
1179	001000	MHS= 1000	;MULTIPLE HEAD SELECT (BIT #9)
1180	002000	NHS= 2000	;NO HEAD SELECTION (BIT #10)
1181	004000	IXE= 4000	;INDEX ERROR (BIT #11)
1182	010000	VU30= 10000	;30VOLT UNSAFE (BIT #12)
1183	020000	PLU= 20000	;PLO UNSAFE (BIT #13)
1184	100000	ACU= 100000	;ACUNSAFE (BIT #15)
1185			
1186			
1187	000001	OF25= 1	;OFFSET 25 MICRO INCHES (BIT #0)
1188	000002	OF50= 2	;OFFSET 50 MICRO INCHES (BIT #1)
1189	000004	OF100= 4	;OFFSET 100 MICRO INCHES (BIT #2)
1190	000010	OF200= 10	;OFFSET 200 MICRO INCHES (BIT #3)
1191	000020	OF400= 20	;OFFSET 400 MICRO INCHES (BIT #4)
1192	000040	OF800= 40	;OFFSET 800 MICRO INCHES (BIT #5)
1193			
1194	000200	OFREV= 200	;OFFSET NEGATIVE (REVERSE) (BIT #5)
1195	002000	HCI= 2000	;HEADER COMPARE INHIBIT (BIT #10)
1196	004000	ECI= 4000	;ERROR CORRECTION CODE INHIBIT (BIT #11)
1197	010000	FMT22= 10000	;FORMAT BIT (BIT #12)
1198			
1199			
1200			
1201			
1202			
1203			
1204			
1205			
1206			
1207			
1208			
1209			
1210			
1211			
1212			
1213			

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

;OFFSET REGISTER (RHOF) (#11)

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

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

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

1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228

000001
000002
000010
000020
000040
000100
040000
100000

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

PSU=	1	;PACK SPEED UNSAFE (BIT #0)
VUF=	2	;VELOCITY UNSAFE (BIT #1)
UWR=	10	;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
PRE=	20	;DISK PACK ROTATION ERROR (BIT #4)
ACL=	40	;AC LOW (BIT #5)
DCL=	100	;DC LOW (BIT #6)
SKI=	40000	;SEEK INCOMPLETE (BIT #14)
OCYL=	100000	;OFF CYLINDER (BIT #15)

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

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

.SBTTL REGISTER ADDRESSES

1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281

002170 176722
 002172 176702
 002174 176704
 002176 176710

 002200 176700
 002202 176714
 002204 176706
 002206 176740
 002210 176732
 002212 176734
 002214 176742
 002216 176716
 002220 176724
 002222 176712
 002224 176726
 002226 176730
 002230 176744
 002232 176746
 002234 176736
 002236 176720

 002240 176750
 002242 176752

 002244 172540
 002246 172542
 002250 172544

;RPO4/5/6 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER

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

;RPO4/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)

RHCS1: 176700 ;CONTROL AND STATUS 1
 RHER1: 176714 ;ERROR #1
 RHDST: 176706 ;DESIRED SECTOR/TRACK ADDRESS
 RHER2: 176740 ;ERROR #2
 RHOF: 176732 ;OFFSET
 RHCA: 176734 ;DESIRED CYLINDER ADDRESS
 RHER3: 176742 ;ERROR #3
 RHAS: 176716 ;ATTENTION SUMMARY
 RHMR: 176724 ;MAINTAINABILITY
 RHDS1: 176712 ;DRIVE STATUS
 RHDT: 176726 ;DRIVE TYPE
 RHSN: 176730 ;SERIAL NUMBER
 RHEC1: 176744 ;ECC POSITION
 RHEC2: 176746 ;ECC PATTERN
 RHCC: 176736 ;CURRENT CYLINDER ADDRESS
 RHLA: 176720 ;LOOK-AHEAD

;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

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

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

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

1282
1283
1284
1285
1286
1287
1288
1289 002252 000000
1290 002254 000000
1291 002256 000000
1292 002260 000000
1293
1294
1295 002262 000000
1296 002264 000000
1297 002266 000000
1298 002270 000000
1299 002272 000000
1300 002274 000000
1301 002276 000000
1302 002300 000000
1303 002302 000000
1304 002304 000000
1305 002306 000000
1306 002310 000000
1307 002312 000000
1308 002314 000000
1309 002316 000000
1310 002320 000000
1311
1312
1313

: THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
: 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

DB:	0	: DATA BUFFER
WC:	C	: WORD COUNT
BA:	0	: BUS ADDRESS
CS2:	0	: CONTROL AND STATUS 2
CS1:	0	: CONTROL AND STATUS 1
ER1:	0	: ERROR #1
DST:	0	: DESIRED SECTOR/TRACK ADDRESS
ER2:	0	: ERROR #2
OF:	0	: OFFSET
CA:	0	: DESIRED CYLINDER ADDRESS
ER3:	0	: ERROR #3
AS:	0	: ATTENTION SUMMARY
MR:	0	: MAINTAINABILITY
DS1:	0	: DRIVE STATUS
DT:	0	: DRIVE TYPE
SN:	0	: SERIAL NUMBER
EC1:	0	: ECC POSITION
EC2:	0	: ECC PATTERN
CC:	0	: CURRENT CYLINDER ADDRESS
LA:	0	: LOOK-AHEAD

:RESERVED LOCATIONS FOR FLAGS AND INTERNAL PROGRAM CONTROL WORDS

004500 000000
004501 000000
004502 000000
004503 000000
004504 000000
004505 000000
004506 000000

REGADR: 0
ERWORD: 0
TSTNM: 0
RP4VEC: 0

:SAVE REGISTER ADDRESS HERE
:SAVE ERROR WORD NUMBER HERE
:TEST NUMBER
:CONTAINS ADDRESS OF LOCATION
:WHERE AN RPO4 INTERRUPT IS TO VECTOR TO
:THIS MUST BE MOVED INTO 'RPVEC' TO BE
:EFFECTIVE.

004507 000000

OFSTVL: 0

:OFFSET VALUE USED IN OFFSET TEST

004512 000024
004513 000000
004514 000000

SAVERE: .BLKW 20.
FINALA: 0
FINACC: 0

:BLOCK TO SAVE REGISTERS
:SAVE LOOK AHEAD REGISTER AT END OF OPERATION
:SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION

:TABLE FOR ATTENTION BITS
:ATTENTION TABLE

004566 001 002 004
004567 010 020 040
004568 100 200

ATABLE: .BYTE 1,2,4,10,20,40,100,200

:RESERVED LOCATIONS FOR UNIT SELECT

004576 000010
004577 000000
004578 000000
004622 000000
004624 000000
004626 000000
004630 000000

UNITS: .BLKW 8.
UNIT: .WORD 0
NOUNIT: .WORD 0
NUNIT: .WORD 0
NOPUSH: 0
SELECT: .WORD 0
UNITSL: .WORD 0

:THIS IS FILLED WITH -1
:UNIT UNDER TEST
:NUMBER OF UNITS PRESENT
:USED TO KEEP TRACK OF UNIT UNDER TEST
:USED TO DETERMIN IF THERE ARE MORE
:THAN ONE UNIT
:ALL ONES INDICATE NONE OF THE OPERATOR
:INTERVENTION TESTS WILL BE PERFORMED
:ALL ONES INDICATE UNIT TO BE SELECTED
:UNIT NO. SELECTED

004632 000000

ERFLGS: 0

:ERROR FLAG

004634 000000

FIRST: 0

:IF ZERO WILL TYPE HEADER
:IF ONES WILL NOT TYPE HEADER

004636 000000

RPO6: 0

:DEVICE TYPE FLAG

004640 000000

RH70: 0

:IF 1. PROGRAM IS RUNNING ON AN RH70
:IF 0. PROGRAM IS ON AN RH11

004642	000000
004644	000000
004646	000000
004650	000000
004652	000000
004654	000000
004656	000000

RUNCTR:	.WORD	0
ATTENT:	0	
TOTALAT:		0
TMP0:	.WORD	0
TMP1:	.WORD	0
TMP4:	.WORD	0
TMP5:	.WORD	0

: 'RUN' LINE DELAY COUNTER TO BE USED
: WHILE THE SILO IS FILLING

: ATTENTION BIT FOR PRESENT UNIT
: TOTAL ATTENTION BITS

: TEMP STORAGE

: TEMP STORAGE

: TEMP STORAGE

E07

MNDEC-11-DZRIJ-A, RFO4 5/6 FUNCT. CONT. DZRIJA.P11

TST-PT 2 ;TEMP STORAGE

MAY11 27(655) 30-MAR-76 18:59 PAGE 33

SEG 0081

1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500

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

.SBTTL SETUP TESTS

```

004660 012737 177777 004624 BEGIN1: MOV    #-1, @#NOPUSH    ; JUMP OVER OPERATOR REQUIRED TESTS
004666 005037 004626          CLR    @#SELECT      ; DO NOT SELECT UNIT
004672 000412          BR     START
004674 012737 177777 004626 BEGIN2: MOV    #-1, @#SELECT      ; SELECT UNIT
004702 005037 004624          CLR    @#NOPUSH    ; DO NOT JUMP OVER ANY TEST
004706 000404          BR     START
004710 005037 004626 BEGIN:   CLR    @#SELECT      ; DO NOT SELECT UNIT
004714 005037 004624          CLR    @#NOPUSH    ; DO NOT JUMP OVER ANY TEST
                                ; NORMAL RUN

```

004720
004720 000005

```

START:
.SBTTL RESET
.SBTTL INITIALIZE THE COMMON TAGS
;; CLEAR THE COMMON TAGS ($CMTAG) AREA
MOV    # $CMTAG, R6    ; FIRST LOCATION TO BE CLEARED
CLR    (R6)+           ; CLEAR MEMORY LOCATION
CMP    # SWR, R6      ; DONE?
BNE    -6             ; LOOP BACK IF NO
MOV    # STACK, SP    ; SETUP THE STACK POINTER
;; INITIALIZE A FEW VECTORS
MOV    # $SCOPE, @#IOTVEC ; IOT VECTOR FOR SCOPE ROUTINE
MOV    # 340, @#IOTVEC+2 ; LEVEL 7
MOV    # $ERROR, @#EMTVEC ; EMT VECTOR FOR ERROR ROUTINE
MOV    # 340, @#EMTVEC+2 ; LEVEL 7
MOV    # $TRAP, @#TRAPVEC ; TRAP VECTOR FOR TRAP CALLS
MOV    # 340, @#TRAPVEC+2 ; LEVEL 7
MOV    # $PWARN, @#PWAVEC ; POWER FAILURE VECTOR
MOV    # 340, @#PWAVEC+2 ; LEVEL 7
CLR    $TIMES         ; INITIALIZE NUMBER OF ITERATIONS
CLR    $ESCAPE        ; CLEAR THE ESCAPE ON ERROR ADDRESS
MOVB  # 1, $ERMAX     ; ALLOW ONE ERROR PER TEST
MOV    # ., $LADR     ; INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV    # ., $LPERR    ; SETUP THE ERROR LOOP ADDRESS
;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV    @#ERRVEC, -(SP) ; SAVE ERROR VECTOR
MOV    # 64$, @#ERRVEC ; SET UP ERROR VECTOR
MOV    # $SWR, SWR    ; SETUP FOR A HARDWARE SWITCH REGISTER
MOV    # $DISP, DISPLAY ; AND A HARDWARE DISPLAY REGISTER
CMP    #-1, @#SWR    ; TRY TO REFERENCE HARDWARE SWR
BNE    66$           ; BRANCH IF NO TIMEOUT TRAP OCCURRED
                                ; AND THE HARDWARE SWR IS NOT = -1
BR     65$          ; BRANCH IF NO TIMEOUT
64$:  MOV    # 65$, (SP) ; SET UP FOR TRAP RETURN
65$:  MOV    # SWREG, SWR ; POINT TO SOFTWARE SWR

```

```

005054 013746 000004  

005060 012737 005114 000004  

005066 012737 177570 001140  

005074 012737 177570 001142  

005102 022777 177777 174030  

005110 001012  

005112 000403  

005114 012737 005122 64$:  

005120 000002 RTI  

005122 012737 000176 001140 65$: MOV

```

```

146 005130 012737 000174 001142      MOV      #DISPREG,DISPLAY
147 005136 012637 000004      66$:    MOV      (SP)+,2#ERRVEC ;;RESTORE ERROR VECTOR

148 005142 012737 000000 177775      MOV      #0,PS ;;SET PROCESSOR STATUS TO 0
149 005150 012737 000200 000036      MOV      #200,2#TRAPVEC+2 ;;TRAP PRIORITY = 4
150 005156 013700 002166      MOV      2#RPVEC,RO ;;GET RP VECTOR ADDRESS
151 005162 012720 036000      MOV      #RPVECT,(RO)+ ;;THIS IS FOR UNTIMELY INTERRUPTS
152 005166 012720 000340      MOV      #340,(RO) ;;RPO4 INTERRUPT SERVICE ROUTINE
153                                     ;;PRIORITY = 7

154 005172 004737 037026      JSR      PC,2#STKINT ;;INITIALIZE THE TTY KEYBOARD
155 005176 005737 004634      TST      2#FIRST ;;IS THIS FIRST TIME ROUND ?
156 005202 001001      BNE      1$ ;;DO NOT GIVE HEADER IF NOT
157 005204 003402      BR       2$ ;;GIVE HEADER IF SO
158 005206 000137 005774      1$:     JMP      2#SND1 ;;SKIP OVERALL PROGRAM HEADER

159                                     2$:
160 005212 104400 005220      TYPE     68$ ;;TYPE ASCIZ STRING
161 005216 000435      BR       67$ ;;GET OVER THE ASCIZ
162 67$:    .ASCIZ <15><12>'RPO4 5/6 FUNCTIONAL CONTROLLER TEST, PART II - DZRJJ-A?'

163 005312 104400 005320      TYPE     70$ ;;TYPE ASCIZ STRING
164 005316 000414      BR       69$ ;;GET OVER THE ASCIZ
165 69$:    .ASCIZ <15><12>'REVISION 21-MAR-76/<15><12>'

166 005350 104400 005356      TYPE     72$ ;;TYPE ASCIZ STRING
167 005354 000433      BR       71$ ;;GET OVER THE ASCIZ
168 71$:    .ASCIZ <15><12>'ALL DCL'S UNDER TEST MUST BE LOCKED ON CORRECT PORT/'

169 005444 104400 005452      TYPE     74$ ;;TYPE ASCIZ STRING
170 005444 000433      BR       73$ ;;GET OVER THE ASCIZ
171 73$:    .ASCIZ <15><12>'IF CHANGES ARE REQUIRED ON PORT SWITCH, A CYCLE UP/'

172 005540 104400 005546      TYPE     76$ ;;TYPE ASCIZ STRING
173 005544 000436      BR       75$ ;;GET OVER THE ASCIZ
174 75$:    .ASCIZ <15><12>'SEQUENCE IS REQUIRED FOR STROBING THE PORT SELECT FLOP/<15><12>'

175 005642 104400 005650      TYPE     78$ ;;TYPE ASCIZ STRING
176 005646 000430      BR       77$ ;;GET OVER THE ASCIZ
177 77$:    .ASCIZ <15><12>'ALL DCL'S NOT UNDER TEST MUST BE SWITCHED OFF'

178 005730 104400 005736      TYPE     80$ ;;TYPE ASCIZ STRING
179 005734 000417      BR       79$ ;;GET OVER THE ASCIZ
180 79$:    .ASCIZ <15><12>'OR LOCKED ON THE OTHER PORT/'

181 005774 012737 177777 004634      SND1:   MOV      #-1,2#FIRST ;;NEXT TIME DO NOT GIVE HEADER

182 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
183 006002 005737 000042      TST      2#42 ;;ARE WE RUNNING UNDER XXDP ACT?

```

1520	006006	001006				BNE	64\$::BRANCH IF YES
1521	006010	023727	001140	000176		CMP	SWR, #SWREG	::SOFTWARE SWITCH REG SELECTED?
1522	006016	001005				BNE	65\$::BRANCH IF NO
1523	006020	104405				GTSWR		::GET SOFT-SWR SETTINGS
1524	006022	000403				BR	65\$	
1525	006024	112737	000001	001134	64\$:	MOVB	#1, \$AUTOB	::SET AUTO-MODE INDICATOR
1526	006032				65\$:			
1528	006032	032777	010000	173100	RH70CK:	BIT	#SW12, 3SWR	::LOOK TO SEE IF USING RH70
1529	006040	001403				BEQ	3\$::IF SW12 = 0, SKIP NEXT
1530	006042	012737	177777	004640		MOV	#-1, 3#RH70	::IF SW12 = 1, CU IS AN RH70
1531	006050				3\$:			

[Handwritten mark]

[Handwritten mark]

[Handwritten mark]

[Handwritten mark]

[Handwritten mark]

150000
150001
150002
150003
150004
150005
150006
150007
150008
150009
150010
150011
150012
150013
150014
150015
150016
150017
150018
150019
150020
150021
150022
150023
150024
150025
150026
150027
150028
150029
150030
150031
150032
150033
150034
150035
150036
150037
150038
150039
150040
150041
150042
150043
150044
150045
150046
150047
150048
150049
150050

```

*****
: *IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM ?
: *IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
: *IF SO MAKE RPO4 INTERRUPTS GO TO 'TIME 1'
: *IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
: *IF NOT MAKE RPO4 INTERRUPTS GO TO 'TIME 2'

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

: *MOV 2#15,2#ERRVEC ;SET TIME-OUT VECTOR

: JMP 2#15 ;DO NOT USE CLOCK
: NOP
: TST 2#PCLCSR ;REFERENCE P-CLOCK STATUS REGISTER
: ;ADDRESS = 172540
: MOV 2#WAIT.P,2#STRPAD+20 ;THERE IS A P-CLOCK
: MOV 2#TIME1,2#RPO4VEC ;THERE IS A P CLOCK SO
: ;VECTOR TO TIME1
: BR 25
15: MOV 2#WAIT.T,2#STRPAD+20 ;THERE IS NO P-CLOCK
*****

```

11
2
4

```

006050 012737 033216 004506 25: MOV 2#TIME2,2#RPO4VEC ;MAKE RPO4/5/6 INTERRUPTS GO TO 'TIME 2'
006056 012737 177777 040434 MOV #-1,2#PRITEM ;CLEAR PREVIOUS ITEM NUMBER

006064 005737 004626 TST 2#SELECT ;WAS IT A 200 START

006077 001442 BEQ TST1 ;BRANCH IF STARTING FROM 200

006072 104400 006100 TYPE 655 ;:TYPE ASCIZ STRING
006076 000424 BR 645 ;:GET OVER THE ASCIZ

:655: .ASCIZ <15><12>/SELECT UNIT NUMBER TO BE TESTED ? <15><12>
645:

006150 104411 RDOCT
006152 042716 177770 BIC #177770,(SP) ;ONLY KEEP LAST 3 BITS
006156 011637 004616 MOV (SP),2#UNIT ;SAVE UNIT TO BE TESTED
006162 012637 004630 MOV (SP)+,2#UNITSL ;SAVE UNIT TO BE TESTED

006166 001403 BEQ TST1 ;BRANCH IF STARTING FROM 200

006170 013737 004630 004616 MOV 2#UNITSL,2#UNIT ;SET UNIT NUMBER

```

1

1615
1616
1617
1618
1619
1620
1621
1622
1623
1624

: *TEST 1 REFERENCE EACH REGISTER

: * REFERENCE EACH REGISTER BY A MOVE INSTRUCTION

```

TST1:  SCOPE
006176 000004
006200 012737 000001 001212      MOV      #1,STIMES      ;;DO 1 ITERATION
006206 012737 000001 004504      MOV      #2-1,#TSTNM   ;THIS SAVES TEST NUMBER
006214 012706 001000                MOV      #STACK, SP    ;SET UP STACK POINTER
006220 012737 040276 000030      MOV      #REGSA1,#EMTVEC;ERROR VECTOR SO THAT
                                ;NO REGISTERS ARE SAVED
006226 012737 006254 000004      MOV      #25,#ERRVEC  ;SET UP FOR BUS TIMEOUT
006234 012700 000024                MOV      #24,R0        ;THERE ARE 24 REG TO TEST
006240 012701 000170                MOV      #R0DB,R1     ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
15:   006244 013102                MOV      2(R1)+,R2    ;READ HARDWARE REG.
006246 005300                DEC      R0           ;COUNT DOWN
006250 001375                BNE     15           ;BRANCH IF 24 NOT DONE
006252 000470                BR      35           ;BRANCH IF 24 DONE
25:   006254 012737 000006 000004      MOV      #ERRVEC+2,#ERRVEC;RESTORE TRAP CATCHER
006262 022626                CMP      (SP)+,(SP)+ ;CLEAN STACK
006264 016137 177776 001200      MOV      -2(R1),STMP1 ;STORE FAILING REG ADDR
006272 104007                ERROR   7            ;REGISTER NON EXISTANT
006274 032777 020000 172636      BIT     #SW13,#SWR   ;INHIBIT ERROR PRINTOUT ?
006302 001052                BNE     45           ;BRANCH IF YES
006304 104400 006312                TYPE    ,65$        ;;TYPE ASCIZ STRING
006310 000424                BR      64$        ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT/
64$:
006362                TYPE    ,67$        ;;TYPE ASCIZ STRING
006362 104400 006370                BR      65$        ;;GET OVER THE ASCIZ
006366 000415                ;;67$: .ASCIZ <15><12>/PROGRAM AND RESTART AT /
66$:
006422                MOV      #BASECH,-(SP) ;GET READY TO TYPE STARTING ADDRESS
                                ;OF "CHANGE OF BASE ADDRESS" ROUTINE
1618 006426 104401                TYPOC
1619 006430 000137 032506      4$:   JMP     #SEOP      ;GO TO END OF PROGRAM -----
1621 006434 012737 040266 000030      3$:   MOV      #ERROR,#EMTVEC;RESTORE ERROR VECTOR
                                ;SO THAT REGISTERS ARE SAVED
1623 006442 012737 000006 000004      MOV      #ERRVEC+2,#ERRVEC;RESTORE TRAP CATCHER

```


16305
16306
16307
16308
16309
16310
16311
16312
16313
16314
16315
16316
16317
16318
16319
16320
16321
16322
16323
16324
16325
16326
16327
16328
16329
16330
16331
16332
16333
16334
16335
16336
16337
16338
16339
16340
16341
16342
16343
16344
16345
16346
16347
16348
16349
16350
16351
16352
16353
16354
16355
16356
16357
16358
16359
16360
16361
16362
16363
16364
16365
16366
16367
16368
16369
16370
16371
16372
16373
16374
16375
16376
16377
16378
16379
16380
16381
16382
16383
16384
16385
16386
16387
16388
16389
16390
16391
16392
16393
16394
16395
16396
16397
16398
16399
16400

*TEST 2 PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT

* CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES

006450 000004
006452 012737 000001 001212
006460 012736 001000

006464 013701 002216
006470 012711 177777
006474 105711

006476 001407

006500 011137 001126
006504 005037 001124
006510 010137 004500
006514 104005

TST2: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;SET STACK POINTER

MOV @RHAS,R1 ;R1 HAS ADDRESS OF RHAS
MOV #-1,@R1 ;WRITE ALL ONES INTO RHAS
TSTB @R1 ;TEST IT FOR ALL 0'S

BEQ TST3 ;BRANCH IF GOOD

MOV @R1,@\$BDDAT ;BAD DATA
CLR @\$GDDAT ;GOOD DATA
MOV R1,@\$REGADR ;FAILING REG. RHAS
ERROR 5 ;RHAS DOES NOT CLEAR BY WRITING ALL
;ONES INTO IT

```

1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662 006516 000004
1663 006520 012737 000001 001212
1664
1665
1666 006526 012737 000003 004504
1667
1668 006534 000005
1669 006536 004737 037026
1670
1671 006542 032777 020000 172370
1672 006550 001026
1673 006552 104400 006560
1674 006556 000423
1675
1676 006626
1677 006626 013701 002216
1678 006632 013702 002176
1679 006636 005012
1680 006640 012700 000010
1681 006644 013704 002202
1682 006650 012714 177777
1683 006654 005212
1684 006656 005300
1685 006660 001373
1686 006662 111137 004646
1687
1688 006666 105037 004647
1689 006672 105711
1690 006674 001402
1691 006676 000137 007260
1692
1693 006702 032777 020000 172230
1694 006710 001402
1695 006712 000137 007616
1696
1697
1698 006716
1699 006716 104400 006724
1700 006722 000421
1701

```

```

*****
*TEST 3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
*
* THE NUMBER OF RPO4/RPO6 DRIVES PRESENT IS FOUND
* BY MOVING ALL ONES TO RHER1 WITH UNIT NUMBER
* IN RHCS2 INCREMENTED FROM ZERO TO SEVEN.
*
* SET BITS IN RHAS WILL GIVE DRIVES PRESENT.
*
* THE DRIVE TYPE IS CHECKED TO BE RPO4 OR RPO6 AND
* UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
*****
TST3: SCOPE
MOV #1,$TIMES ;;DO 1 ITERATION
MOV #4-1,$TSTNM ;THIS SAVES TEST NUMBER
RESET ;START WITH AN INIT
JSR PC,$$TKINT ;INITILIZE THE TTY KEYBOARD
BIT #SW13,$SWR ;INHIBIT ERROR TYPEOUT?
BNE 4$ ;BRANCH IF YES
TYPE ,65$ ;;TYPE ASCIZ STRING
BR 64$ ;;GET OVER THE ASCIZ
65$: .ASCIZ <15><12><15><12>/LOOKING AT RHAS - DRIVES PRESENT/
64$:
4$: MOV 2$RHAS,R1 ;R1 HAS ADDR. OF RHAS
MOV 2$RHCS2,R2 ;R2 HAS ADDR. OF RHCS2
CLR 2$R2 ;CLEAR RHCS2
MOV #9,$R0 ;COUNT
MOV 3$RHER1,R4 ;R4 HAS ADDR. OF RHER1
1$: MOV #-1,$R4 ;MOVE ERRORS INTO RHER1
INC 2$R2 ;INCREMENT UNIT NO.
R0 ;COUNT
BNE 1$ ;BRANCH IF 8 NOT DONE
MOVB 2$R1,$TOTALAT ;SAVE TOTAL ATTENTION
;USED IN DRIVE CLEAR TEST
CLRB 2$TOTALAT+1 ;CLEAR UPPER BYTE
TSTB 2$R1 ;TEST FOR ANY DRIVES PRESENT
BEQ 2$ ;IF NONE THERE - SAY SO
JMP XE2 ;SOME THERE - LOAD TABLE
2$: BIT #SW13,$SWR ;INHIBIT ERROR TYPE OUT
BNE 3$ ;BRANCH IF NO
JMP SELTST ;CHECK FOR SELECTED UNIT START AND LOAD
;"UNITS" TABLE WITH DESIRED DRIVE IF SO
3$:
TYPE ,67$ ;;TYPE ASCIZ STRING
BR 66$ ;;GET OVER THE ASCIZ
67$: .ASCIZ <15><12><15><12>/NO DRIVES PRESENT - RHAS = 0/

```

TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

```

1702 006766 56$:
1703 006766 104400 006774 TYPE 69$ ;;TYPE ASCIZ STRING
1704 006772 000430 BR 68$ ;;GET OVER THE ASCIZ
1705 ;;69$: .ASCIZ <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
1706 007054 68$:
1707 007054 104400 007062 TYPE 71$ ;;TYPE ASCIZ STRING
1708 007060 000430 BR 70$ ;;GET OVER THE ASCIZ
1709 ;;71$: .ASCIZ <15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/
1710 007142 70$:
1711 007142 104400 007150 TYPE 73$ ;;TYPE ASCIZ STRING
1712 007146 000442 BR 72$ ;;GET OVER THE ASCIZ
1713 ;;73$: .ASCIZ <15><12>/TO LOOP ON THIS TEST WO PRINTOUT SET SWITCHES 13,8,1 & 0<15><12
1714 007254 72$:
1715
1716 007254 000137 032506 JMP 2*$EOP ;GO OUT ----->
1717
1718
1719 ;*SET UP THE UNITS TABLE
1720
1721 007260 XE2:
1722 007260 012700 000010 2$: MOV #8, R0 ;COUNTER
1723 007264 012703 004575 MOV #UNITS, R3 ;POINTER
1724 007270 012723 177777 3$: MOV #-1, (R3)+ ;PRESE* BLOCK TO ALL ONES
1725 007274 005300 DEC R0 ;COUNT
1726 007276 001374 BNE 3$ ;BRANCH IF 8 NOT DONE
1727 007300 012703 004575 MOV #UNITS, R3 ;POINTER
1728 007304 005005 CLR R5
1729 007306 005037 004620 CLR 2*#NOUNIT ;NO. OF UNITS PRESENT
1730 007312 012700 000010 MOV #8, R0 ;COUNTER
1731 007316 011137 001176 MOV 2R1, 2*$TMPD ;TEMPORARY STORAGE
1732 007322 006037 001176 4$: ROR 2*$TMPD ;SET CARRY IF ONE IN 0 BIT
1733 007326 103120 BCC 5$ ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
1734
1735 007330 010577 172642 MOV R5, 2RHCS2 ;INSERT UNIT NUMBER INTO UA BITS
1736 007334 022777 024020 172662 CMP #24020, 2RHDT ;IS THIS A DUAL PORT RPO4 ?
1737 007342 001503 BEQ 6$ ;TYPE DRIVE NO. IF SO
1738 007344 022777 020020 172652 CMP #20020, 2RHDT ;IS THIS A SINGLE PORT RPO4 ?
1739 007352 001477 BEQ 6$ ;TYPE NO. IF SO
1740
1741 ;;*****
1742 007354 022777 024021 172642 CMP #24021, 2RHDT ;IS THIS A DUAL PORT RPO5 ?
1743 007362 001473 BEQ 6$ ;TYPE DRIVE NO. IF SO
1744 007364 022777 020021 172632 CMP #20021, 2RHDT ;IS THIS A SINGLE PORT RPO5 ?
1745 007372 001467 BEQ 6$ ;TYPE THE NO. IF SO
1746
1747 007374 022777 024022 172622 CMP #24022, 2RHDT ;IS THIS A DUAL PORT RPO6 ?
1748 007402 001463 BEQ 6$ ;TYPE DRIVE NO. IF SO
1749 007404 022777 020022 172612 CMP #20022, 2RHDT ;IS THIS A SINGLE PORT RPO6 ?
1750 007412 001457 BEQ 6$ ;TYPE DRIVE NO. IF SO
1751 ;;*****
1752
1753
1754 007414 104400 007422 TYPE 65$ ;;TYPE ASCIZ STRING
1755 007420 000410 BR 64$ ;;GET OVER THE ASCIZ

```

M07

```

1756          ;;65$: .ASCIZ <15><12>/UNIT NUMBER /
1757 007442    64$:
1758 007442    010546    MOV     R5,-(SP)      ;GET READY TO TYPE UNIT NUMBER
1759 007444    104404    TYPDS
1760 007446    104400    007454    TYPE     67$        ;;TYPE ASCIZ STRING
1761 007452    C00406    BR      66$        ;;GET OVER THE ASCIZ
1762          ;;67$: .ASCIZ /, RHDT = /
1763 007470    66$:
1764 007470    017746    172530    MOV     @RHDT,-(SP)  ;GET READY TO TYPE RHDT
1765 007474    104401    TYPDC
1766 007476    104400    007504    TYPE     69$        ;;TYPE ASCIZ STRING
1767 007502    C00422    BR      68$        ;;GET OVER THE ASCIZ
1768          ;;69$: .ASCIZ ? - NOT AN RPO4/RPO5/RPO6 DEVICE !!?
1769 007550    69$:
1770 007550    000407    BR      5$         ;NO RPO4/RPO5/RPO6 FOUND SO INCR TABLE
1771
1772 007552    010523    5$:      MOV     R5,(R3)+
1773 007554    104400    001223    TYPE     $CRLF
1774 007560    010546    MOV     R5,-(SP)
1775 007562    104404    TYPDS    ;TYPE DRIVE NO.
1776 007564    005237    004620    INC     @#NUNIT     ;NUMBER OF DRIVES
1777
1778 007570    005205    5$:      INC     R5         ;INCR UNIT NUMBER
1779 007572    005300    DEC     R0         ;DECR NO. OF UNITS LOOKED AT
1780 007574    001252    BNE     4$         ;TEST THE NEXT UNIT
1781
1782 007576    013737    004576    004616    MOV     @#UNITS,@#UNIT ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
1783 007604    013737    004620    004622    MOV     @#NUNIT,@#NUNIT ;SAVE NO. OF UNITS
1784 007612    005337    004622    DEC     @#NUNIT     ;IF NUNIT = 0 THEN ONLY ONE UNIT
1785          ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
1786
1787 007616    005737    004626    SELTST: TST     @#SELECT ;STARTING ADDRESS 200 ?
1788
1789 007622    001403    BEG     TST4       ;BRANCH IF STARTING FROM 200
1790
1791 007624    013737    004630    004616    MOV     @#LNITSL,@#UNIT ;SET UNIT NUMBER
    
```

1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845

007632 000004
007634 012737 000001 001212
007642 012737 010534 0C1106
007650 012706 001000
007654 012737 030004 004504
007662 004737 033004
007666 005037 004644
007672 005737 004616
007676 001107
007700 012700 000041
007704 122710 000011
007710 001102
007712 005737 004626
007716 001006
007720 012700 004576
007724 005720
007726 022710 177777
007732 001065
007734 104400 007742
007740 000434
010032
010032 104400 010040
010036 000421

::*****
:*TEST 4 TYPE SERIAL NUMBER AND DRIVE TYPE

;* SET APPROPRIATE ATTENTION 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

::*****

tST4: SCOPE
MOV #1, \$TIMES ;; DO 1 ITERATION
MOV #1\$, \$LPADR ;; SET SCOPE LOOP ADDRESS
MOV #STACK, SP ; RESET STACK
MOV #4, #TSTNM ; SAVE TEST NUMBER
JSR PC, #CLDISK ; SET R1-RHCS1, R2-RHCS2
; R3-RHDS1, R4-RHER1
; GIVE RH-11 INITIALIZE
; SETUP UNIT NUMBER
CLR #ATTENT ; CLEAR UNIT UNDER TEST ATTENTION
TST #UNIT ; IS THE "UNIT" = 0 ?
BNE 20\$; IF NOT, SKIP NEXT MODS
MOV #41, R0 ; IF SO, CHECK THE LOAD MEDIA LOCATION
CMPB #11, (R0) ; WAS IT AN RPO4/5/6 ?
BNE 20\$; IF NOT, GO AHEAD AND TEST UNIT #0
TST #SELECT ; WAS UNIT #0 SELECTED ?
; (IE. 210 START ?)
BNE 19\$; IF SO, CHANGE PACK

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

MOV #UNITS, R0 ; IF NOT, LOAD THE UNITS TABLE POINTER
TST (R0)+ ; SELECT THE NEXT UNIT IN TABLE
; (DOUBLE INCREMENT THE POINTER, R0)
CMP #-1, (R0) ; IS THERE ANOTHER TABLE ENTRY PRESENT ?
BNE 18\$; IF SO, USE NEXT DRIVE & DECR "NOUNITS"
; IF NOT, CHANGE PACK ON UNIT #0

19\$: TYPE 65\$;; TYPE ASCIZ STRING
BR 64\$;; GET OVER THE ASCIZ
65\$: .ASCIZ <15><12><15><12> DISMOUNT PACK FROM UNIT #0 AND MOUNT A SCRATCH PACK.
64\$: TYPE 67\$;; TYPE ASCIZ STRING
BR 66\$;; GET OVER THE ASCIZ

```

:67S: .ASCIZ <15><12>/PRESS CONTINUE WHEN FINISHED/<15><12>
66S: HALT
BR 20S :CONTINUE, USING SCRATCH PACK ON UNIT #0

16S: MOV .RD),2#UNIT :SET UP NEW UNIT UNDER TEST
DEC 2#NUNITS :DECR BECAUSE UNIT #0 WON'T BE TESTED

20S: MOV 2#UNIT,RD :RD NOW CONTAINS UNIT NO

:*****
: CLR 2#RPC6 :CLEAR RPC6 DEVICE TYPE FLAG
: MOV RD,2#RHC2 :SET UP UNIT ADDRESSING
: CMP #24022,2#RHD :DUAL PORT RPC6 ?
: BEQ 2S :YES...SET FLAG
: CMP #20022,2#RHD :SINGLE PORT RPC6 ?
: BEQ 2S :YES...SET FLAG
: BR 3S :DON'T SET RPC6 FLAG
25: MOV #1,2#RPC6 :SET THE FLAG

35: :ASSUME THE NEXT UNIT IS AN RPC4-RPC5
:*****

: MOVB ATABLE(RD),2#ATTENT :SET APPROPRIATE ATTENTION BIT
: TYPE 69S :TYPE ASCIZ STRING
BR 68S :GET OVER THE ASCIZ
:69S: .ASCIZ <15><12>/TESTING DRIVE NUMBER/
68S: MOV 2#UNIT,-(SP) :UNIT NO. TO STACK
TYPDS :TYPE DRIVE NO.
: TYPE 71S :TYPE ASCIZ STRING
BR 70S :GET OVER THE ASCIZ
:71S: .ASCIZ <15><12>/SERIAL NO. = /
70S: MOV 2#RHSN,-(SP) :SAVE 2#RHSN FOR TYPEOUT
: TYPDC :GO TYPE--OCTAL ASCII(ALL DIGITS)
: TYPE 73S :TYPE ASCIZ STRING
BR 72S :GET OVER THE ASCIZ
:73S: .ASCIZ <15><12>/DRIVE TYPE =
72S: MOV 2#RHDT,-(SP) :SAVE 2#RHDT FOR TYPEOUT
: TYPDC :GO TYPE--OCTAL ASCII(ALL DIGITS)

```

:*TYPE OUT THE DRIVE TYPE IN ASCII

::*****

010324	022777	024020	171672	CMP	#24020,DRHDT	:DUAL PORT RPO4 ?
010332	001424			SEQ	45	:TYPE ASCII MESSAGE OUT
010334	022777	020020	171662	CMP	#20020,DRHDT	:SINGLE PORT RPO4 ?
010342	001420			BEQ	45	:TYPE THE MESSAGE

010354	022777	024021	171652	CMP	#24021,DRHDT	:DUAL PORT RPO5 ?
010356	001433			BEQ	55	:TYPE THE MESSAGE
010357	022777	020021	171642	CMP	#20021,DRHDT	:SINGLE PORT RPO5 ?
010362	001427			BEQ	55	:TYPE THE MESSAGE

010364	022777	024022	171632	CMP	#24022,DRHDT	:DUAL PORT RPO6 ?
010372	001442			BEQ	65	:TYPE THE MESSAGE
010374	022777	020022	171622	CMP	#20022,DRHDT	:SINGLE PORT RPO6 ?
010402	001436			BEQ	65	:TYPE THE MESSAGE

010404				45:		
010405	104400	010412		TYPE	755	::TYPE ASCIZ STRING
010410	000413			BR	745	::GET OVER THE ASCIZ

::755: .ASCIZ <15><12>/DRIVE IS AN RPO4/<15><12>

010440				745:		
010441	000435			BR	15	:SKIP NEXT ONES

010442				55:		
010443	104400	010450		TYPE	775	::TYPE ASCIZ STRING
010446	000413			BR	765	::GET OVER THE ASCIZ

::775: .ASCIZ <15><12>/DRIVE IS AN RPO5/<15><12>

010476				765:		
010476	000416			BR	15	:SKIP NEXT

010500				65:		
010500	104400	010506		TYPE	795	::TYPE ASCIZ STRING
010504	000413			BR	785	::GET OVER THE ASCIZ

::795: .ASCIZ <15><12>/DRIVE IS AN RPO6/<15><12>

010534				785:		
						::*****

010534	005777	171466		15:	TST	DRHDT	:READ SERIAL NO. AND DRIVE TYPE
010540	005777	171460			TST	DRHDT	:THESE TWO ARE TO HELP SCOPE LOOPS
010544	017737	171456	002310		MOV	DRHDT,DRSN	:SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
010552	017737	171446	002306		MOV	DRHDT,DRDT	:SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS

MACY11-22RJJ-A RPO4 5 6 FUNCT. CONT. TST-PT 2
D2RJJ.A.P11 TYPE SERIAL NUMBER AND DRIVE TYPE

1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

*TEST 5 CHECK MOL TO BE HIGH

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

†STS: SCOPE

010560 000004
010562 012737 000005 004504 MOV #6-1.2#TSTNM ;THIS SAVES TEST NUMBER

010570 004737 033004 JSR PC,3#CLDISK ;GIVE INITILIZE
010574 032713 010000 BIT #MOL,DR3 ;CHECK MOL IN RHDS1

010600 001144 BNE TST6 ;BRANCH IF MOL HIGH

010602 104400 010610 TYPE 655 ;:TYPE ASCIZ STRING
010606 000420 BR 645 ;:GET OVER THE ASCIZ
645: .ASCIZ <15><12>/DRIVE IS OFFLINE - MOL IS LOW/

010650 104400 010656 TYPE 675 ;:TYPE ASCIZ STRING
010654 000424 BR 655 ;:GET OVER THE ASCIZ
675: .ASCIZ <15><12>/HIT START ON DRIVE TO GET IT ON LINE

010726 104400 010734 TYPE 695 ;:TYPE ASCIZ STRING
010732 000431 BR 685 ;:GET OVER THE ASCIZ
695: .ASCIZ <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS HIGH/

011016 032713 010000 BIT #MOL,DR3 ;CHECK MOL IN RHDS1
011022 001775 BEQ 15 ;WAIT IF MOL IS STILL LOW

011024 104400 011032 TYPE 715 ;:TYPE ASCIZ STRING
011030 000430 BR 705 ;:GET OVER THE ASCIZ
715: .ASCIZ <15><12>/GOOD - MOL IS HIGH. PROGRAM WILL BE EXECUTED/

011112

1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

*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

TST6: SCOPE

```

011112 000004
011114 012737 000006 004504      MOV      #7-1,2#TSTNM      ;THIS SAVES TEST NUMBER
011122 012706 001000              MOV      #STACK,SP        ;RESET STACK
011126 004737 033004              JSR      PC,2#CLDISK      ;SET R1-RHCS1, R2-RHCS2
                                ;R3-RHDS1, R4-RHER1
                                ;GIVE RH-!1 INITIALIZE
                                ;SETUP UNIT NUMBER
011132 013700 002166              MOV      2#RPVEC,RO       ;GET RP VECTOR ADDRESS
011136 012720 011204              MOV      #RPTRP!,(RO)+   ;THIS IS FOR TIMELY INTERRUPTS
011142 012710 000340              MOV      #340,(RO)       ;RPO4 INTERRUPT SERVICE ROUTINE
                                ;PRIORITY = 7
011146 012737 000200 177776      MOV      #200,PS         ;SET PROCESSOR PRIORITY 2 4
011154 012711 000300              MOV      #RDY!IE,2R1     ;RDY, IE IN RHCS1 SHOULD CAUSE INTERRUPT
011160 013737 033502 001200      MOV      2#TIMCNT,2#STMP1;COUNTER
011166 005337 001200      IS:    DEC      2#STMP1    ;WAIT FOR INTERRUPT
011172 001375              BNE     IS              ;BRANCH IF NOT ZERO
                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
011174 104065              ERROR   65              ;INTERRUPT DID NOT OCCUR
011176 012712 000040              MOV      #CLR,2R2        ;CLEAR CONTROLLER VIA CS2
011202 000407              BR      TST7            ;BRANCH TO NEXT TEST -----)
011204 022626              RPTRP1: CMP      (SP)+,(SP)+ ;RESTORE STACK
011206 022711 004200              CMP      #DVA!RDY,2R1   ;IE SHOULD BE LOW
011212 001403              BEQ     TST7            ;CONTINUE IF GOOD -----)
011214 104065              ERROR   65              ;INTERRUPT OCCURED BUT
                                ;IE FAILED TO RESET
011216 012712 000040              MOV      #CLR,2R2        ;CLEAR CONTROLLER

```

F08

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070

*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 INTERRUPT

TST7: SCOPE

011222 000004

011224 012737 000007 004504

MOV #10-1, @TSTNM ;THIS SAVES TEST NUMBER

011232 012706 001000

MOV #STACK, SP ;RESET STACK

011236 004737 033004

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

011242 013700 002166

MOV @RPVEC, RD ;GET RP VECTOR ADDRESS

011246 012720 011312

MOV #RPTR2, (RD)+ ;THIS IS FOR UNTIMELY INTERRUPTS

011252 012710 000340

MOV #340, (RD) ;RPO4 INTERRUPT SERVICE ROUTINE

011256 012737 000240 177776

MOV #240, PS ;SET PROCESSOR PRIORITY = 5

011264 012711 000300

MOV #RDY!IE, @R1 ;RDY, IE IN RHSC1 WOULD CAUSE INTERRUPT

011270 013737 033502 001200

MOV @TIMCNT, @STMP1 ;COUNTER

011276 005337 001200

15: DEC @STMP1 ;WAIT FOR INTERRUPT

011302 001375

BNE 15 ;BRANCH IF NOT ZERO

011304 012712 000040

MOV #CLR, @R2 ;BEFORE THIS IS ZERO INTERRUPT WOULD OCCUR
;CLEAR CONTROLLER

011310 000404

BR TST10 ;NO INTERRUPT SO CONTINUE -----)

011312 022626

RPTR2: CMP (SP)+, (SP)+ ;RESTORE STACK

011314 104065

ERROR 65 ;INTERRUPT OCCURRED WITH
;PROCESSOR PRIORITY SAME AS DISK

011316 012712 000040

MOV #CLR, @R2 ;CLEAR CONTROLLER

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

:TEST 10 SET VV BIT #6 IN RHDS1

:* THIS TEST SETS VV IN RHDS!
:* THERE IS A RESET AT THE BEGINING OF THE TEST
:* FOR ERROR RECOVERY ONLY.

†ST10: SCOPE

011322 000004

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

011324 000005

RESET

011326 004737 037026
011332 012737 000000 177776

JSR PC,0#STKINT ;INITILIZE TK
MOV #0,PS

011340 012706 001000
011344 012737 000010 004504

MOV #STACK,SP ;RESET STACK
MOV #10,0#†STNM ;SAVE TEST NUMBER

011352 004737 033004

JSR PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

011356 004737 033042
011362 104400 056662
011366 000000

JSR PC,0#CHECK ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
TYPE .CPHALT ;CANNOT CONTINUE TESTS IF THEY DON'T
HALT ;STOP TESTING

011370 013777 002360 170602

MOV 0#PKACK,0RHCS1 ;GET READY FOR PKACK
;PACK ACKNOWLEDGE WITH 22 IN RHCS1

;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE

011376 004037 033152
011402 002172
011404 004512

JSR RC,0#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRES OF WHERE
;THE REGISTERS ARE SAVED
18. ;NUMBER OF REGISTERS
;SAVED = 18.

011406 000022

011410 013777 004506 170550

MOV 0#RPHVEC,0RPVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS

011416 013746 002360
011422 052716 000001

MOV 0#PKACK, -(SP) ;GET READY TO MOVE COMMAND
BIS #GO,0# ;GET READY TO SET GO
;WITHOUT INTERRUPT ENABLE

011426 012677 170546

MOV (SP)+,0RHCS1 ;GO WITH
;22 IN RHCS1 FOR PACK ACKNOWLEDGE

```

11432      011432 011100      MOV      R1,R0      ;WITH INTERRUPT DISABLED
11433      011433 011305      MOV      R3,R5      ;SAVE RHCS1 DURING ABOVE OPERATION
11434      011434 011305      MOV      R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION
11435
11436      011436 104412      WAT              ;WAIT FOR VV BIT TO SET
11437      011437 002222      RHDS1           ;WAIT FOR RHDS1 REGISTER
11438      011438 000100      VV              ;WAIT FOR VV BIT IN RHDS1 REGISTER
11439      011439 000001      I.              ;ALLOW 10 MICRO SECONDS
11440      011440 000001      I.              ;VV MUST SET BETWEEN
11441      011441 000001      I.              ;00 AND 20 MICRO SECONDS
11442
11443      ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
11444      ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
11445
11446
11447
11448
11449
11450      011450 013746 002360      MOV      R0,R#PKACK,-(SP) ;SAVE COMMAND
11451      011451 052716 004200      BIS      R0,R#DVA!RDY,(SP) ;INCLUDE DVA!RDY
11452      011452 011637 001124      MOV      (SP),R#SGDDAT ;SAVE FOR PRINTOUT
11453      011453 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!RDY
11454      ;AND COMMAND SHOULD BE SET
11455      011455 001405      BEQ      64$ ;BRANCH IF GOOD
11456      011456 010037 001126      MOV      R0,R#SBDDAT ;BAD DATA
11457      011457 010137 004500      MOV      R1,R#REGADR ;FAILING REGISTER RHCS1
11458      011500 104021      ERROR    21 ;DURING ABOVE OPERATION ONLY
11459      ;COMMAND AND DVA!RDY SHOULD BE SET
11460      011502 012746 010700      64$: MOV      R0,R#MOL!DPR!DRY!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
11461      011506 011637 001124      MOV      (SP),R#SGDDAT ;SAVE FOR PRINTOUT
11462      011512 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!DRY!VV
11463      ;SHOULD BE SET
11464      011514 001405      BEQ      66$ ;BRANCH IF GOOD
11465      011516 010537 001126      MOV      R5,R#SBDDAT ;BAD DATA
11466      011522 010337 004500      MOV      R3,R#REGADR ;FAILING REGISTER RHDS1
11467      011526 104063      ERROR    63 ;DURING ABOVE OPERATION ONLY
11468      ;MOL!DPR!DRY!VV SHOULD BE SET
11469
11470      011530      66$:
11471      011530 004037 033674      JSR      R0,R#CHREG ;CHANGE BITS IN SAVED REGISTER
11472      011534 002222      RHDS1 ;CHANGE RHDS1 REGISTER
11473
11474      011536 000001      I. ;1 BIT/BITS TO BE CHANGED
11475      011540 000001      I. ;NEW VALUE OF VV IS 1
11476      011542 000100      VV ;CHANGE VV BIT
11477
11478      ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
11479      ;*WITH AFTER PACK ACKNOWLEDGE
11480
11481
11482
11483
11484      011544 004037 034002      JSR      R0,R#COMREG ;COMPARE SAVED REGISTERS WITH
11485      ;PRESENT VALUE
11486      011550 004512      SAVERE ;GOOD DATA SAVED IN 'SAVERE'
11487      011552 002254      WC ;TEST DATA STARTING FROM 'RHWC'
11488      011554 000022      18. ;18. REGISTERS TO BE COMPARED

```


.SBTTL DATA TRANSFER RELATED ERRORS (USING MEDIA)

*TEST 11 LAST BLOCK TRANSFERED-RHDS1 LBT

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

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

†ST11: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #11, @†STNM ;SAVE TEST NUMBER
JSR PC, @*CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

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

TST @*RPO6 ;TEST FOR RPO6 DRIVE
BEQ 11\$;TREAT UNIT AS AN RPO4
;TREAT UNIT AS AN RPO6

JSR RC, @*FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
5 ;NUMBER OF WORDS SAVED
11456 ;FIRST DATA WORD
<18.*400>!<21.> ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
<26.*2000>!<18.*40>!<21.> ;FIFTH DATA WORD
BR 12\$;CONTINUE WITH SET UP

11\$:

JSR RC, @*FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
5 ;NUMBER OF WORDS SAVED
10632 ;FIRST DATA WORD
<18.*400>!<21.> ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
<26.*2000>!<18.*40>!<21.> ;FIFTH DATA WORD

2198
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252

011566 000004
011570 012706 001000
011574 012737 000011 004504
011602 004737 032004
011606 005737 004636
011612 001412
011614 004037 032630
011620 002370
011622 000005
011624 011456
011626 011025
011630 000000
011632 000000
011634 065125
011636 000411
011640
011640 004037 032630
011644 002370
011646 000005
011650 010632
011652 011025
011654 000000
011656 000000
011660 065125

K08

MNDEC-11-DZRJJ-A, RPO4, 5/6 FUNCT. CONT. TST-PT 2
DZRJJA.P11 †11 LAST BLOCK TRANSFERED-RHDS1 LBT

MACY11 27(655) 30-MAR-76 18:59 PAGE 52

SEG 0100

2253 011662

125:

:*FILL READ INTO BUFFER WITH ALL ONES

2258 011662 004037 032654

JSR RO.2#CLAREA :CLEAR 256 WORDS FROM REINTO

2259 011666 003434

REINTO :STARTING FROM REINTO

2260 011670 000256

256 :256 WORDS

2261 011672 177777

-1 :FILL WITH -1

2262
2263
2264

Handwritten signature or initials

```

2264
2265
2266
2267
2268
2269 011674 005737 004636
2270 011700 001412
2271
2272
2273
2274 011702 004037 034766
2275 011706 001456
2276 011710 025
2277 011711 022
2278 011712 177773
2279
2280 011714 002370
2281
2282
2283 011716 000000
2284 011720 010000
2285
2286
2287 011722 002344
2288
2289
2290 011724 000411
2291
2292 011726
2293
2294 011726 004037 034766
2295 011732 000632
2296 011734 025
2297 011735 022
2298 011736 177773
2299
2300 011740 002370
2301
2302
2303 011742 000000
2304 011744 010000
2305
2306
2307 011746 002344
2308
2309
2310 011750
2311
2312
2313
2314
2315 011750 004037 033152
2316 011754 002172
2317 011756 004512

```

```

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

; *****
; TST 2#RPO6 ; TEST FOR RPO6 DRIVE
; SEQ 7$ ; TREAT UNIT AS RPO4
; *****

JSR RO,2#RUN ; SETUP TO RUN FOR DATA COMMAND
814. ; CYLINDER 814.
.BYTE 21. ; SECTOR 21.
.BYTE 18. ; TRACK 18.
-1-4 ; WORD COUNT (DATA) = 1 +
; 4 HEADER WORDS
WRFROM ; BUS ADDRESS
; STARTING ADDRESS OF DATA
; BUFFER = WRFROM
0 ; DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ; 16 BITS PER WORD FORMAT
; DO NOT INHIBIT ECC CORRECTION
; DO NOT INHIBIT HEADER COMPARE
WRIFOR ; GET READY TO DO A WRIFOR
; WRITE HEADER AND DATA WITH 62 IN RHCS1

BR 8$ ; CONTINUE WITH TEST

7$:

JSR RO,2#RUN ; SETUP TO RUN FOR DATA COMMAND
410. ; CYLINDER 410.
.BYTE 21. ; SECTOR 21.
.BYTE 18. ; TRACK 18.
-1-4 ; WORD COUNT (DATA) = 1 +
; 4 HEADER WORDS
WRFROM ; BUS ADDRESS
; STARTING ADDRESS OF DATA
; BUFFER = WRFROM
0 ; DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ; 16 BITS PER WORD FORMAT
; DO NOT INHIBIT ECC CORRECTION
; DO NOT INHIBIT HEADER COMPARE
WRIFOR ; GET READY TO DO A WRIFOR
; WRITE HEADER AND DATA WITH 62 IN RHCS1

8$:

; *NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE

JSR RO,2#SAVER ; SAVE REGISTERS
RHWC ; RHWC IS THE FIRST REGISTER SAVED
SAVERE ; STARTING ADDRESS OF WHERE

```



```

012074 000001          1          ;NEW VALUE OF BIT0 IS 1
012076 000001          BIT0        ;CHANGE BIT0 BIT
012100 004037 032706   JSR      RO,0#FILLRE ;MOV 0 INTO SAVED RHDST
012104 002204          RHDST       ;SAVED REGISTER TO CHANGE
012106 000000          0           ;DATA
                                     ;*COMPARE ALL REGISTERS
012110 004037 034002   JSR      RO,0#COMREG ;COMPARE SAVED REGISTERS WITH
                                     ;PRESENT VALUE
012114 004512          SAVERE       ;GOOD DATA SAVED IN 'SAVERE'
012116 002254          WC           ;TEST DATA STARTING FROM 'RHWC'
012120 000021          17.         ;17. REGISTERS TO BE COMPARED
012122 012126          1$          ;RETURN TO 1$ ON ERROR
012124 012132          2$          ;RETURN TO 2$ ON NO ERROR
012126 104045          1$:        ERRJR   45 ;WRITING ON THE LAST BLOCK
012130 000207          RTS      PC    45 ;IE. CYLINDER 410./814., SECTOR 21
                                     ;TRACK 18 CAUSED
                                     ;IMPROPER REGISTER CHANGE
                                     ;GOOD DATA GIVES WHAT
                                     ;SHOULD BE THERE
                                     ;RECEIVED DATA GIVES WHAT
                                     ;WAS THERE AFTER WRITE
                                     ;ON LAST BLOCK

```

;*NOW A READ DATA WILL BE DONE ON SAME CYLINDER, SECTOR & TRACK
;*CLEAR ANY PREVIOUS ERRORS

28:

012126
012128 004737 033004

JSR RC,2#C_DISK :SET R1-RHCS1, R2-RHCS2
:R3-RHOS1, R4-RHER1
:GIVE RH-11 INITIALIZE
:SETUP UNIT NUMBER

;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

012129 004737 033004
012131 004737 033004
012133 004737 033004
012135 004737 033004

JSR RC,2#FLHEAD :SAVE HEADER DATA IN WRFROM
WRFROM :LOCATION WHERE SAVED
1 :NUMBER OF WORDS SAVED
26.*2000!(18.*40)!(21.) ;FIRST DATA WORD

012137 004737 033004
012139 004737 033004
012141 004737 033004
012143 004737 033004

JSR RC,2#CLAREA :CLEAR 256 WORDS FROM WRFROM+2
WRFROM+2 :STARTING FROM WRFROM+2
256. :256 WORDS
0 :FILL WITH 0

;*FIRST THE DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
;*READ COMMAND IS LOADED

012145 005737 034636
012147 005737 034636

::*****
TST 2#RPO6 ;TEST FOR RPO6 DRIVE
REQ 95 ;TREAT UNIT AS RPO4
::*****

012149 004737 034756
012151 004737 034756
012153 004737 034756
012155 004737 034756
012157 004737 034756
012159 004737 034756
012161 004737 034756
012163 004737 034756

JSR RC,2#RUN :SETUP TO RUN FOR DATA COMMAND
814. :CYLINDER 814.
.BYTE 21. :SECTOR 21.
.BYTE 18. :TRACK 18.
-256. :WORD COUNT = 256.
REINTO :BUS ADDRESS
:STARTING ADDRESS OF DATA
0 :BUFFER = REINTO
FMT22 :DO NOT INHIBIT BUS ADDRESS INCREMENT
:16 BITS PER WORD FORMAT
:DO NOT INHIBIT ECC CORRECTION
:DO NOT INHIBIT HEADER COMPARE
READAT :GET READY TO DO A READAT
:READ DATA WITH 70 IN RHCS1

012165 000000
012167 010000

012169 002346

012171 000411

95:

012173 004027 03-766
012175 004027 03-766

JSR RC,2#RUN :SETUP TO RUN FOR DATA COMMAND

5 6 FUNC. CONT. TEST-PT 2
LAST BLOCK TRANSFERED-RHCSI LBT

012234 002234
012236 002236
012236 004037 033152
012244 004512
012246 000222
012250 004727 033064
012254 104400 056662
012260 000200
012262 013777 004506 167676

012270 013746 002346
012274 052716 000101
012300 012677 167674

012304 104412
012308 000200
012312 000200
012316 001614
012318 001602

012316 004037 032706

410.
:BYTE 21:
:BYTE 18:
-256.
REINTO

J
FMT22

READAT

:CYLINDER 410.
:SECTOR 21.
:TRACK 18.
:WORD COUNT = 256.
:BUS ADDRESS
:STARTING ADDRESS OF DATA
:BUFFER = REINTO
:DO NOT INHIBIT BUS ADDRESS INCREMENT
:16 BITS PER WORD FORMAT
:DO NOT INHIBIT ECC CORRECTION
:DO NOT INHIBIT HEADER COMPARE
:GET READY TO DO A READAT
:READ DATA WITH 70 IN RHCSI

105:

:*SAVE REGISTERS FOR COMPARISON AFTER READ DATA
JSR RO,0#SAVER
RHWCSAVERE

18.

JSR PC,0#CHECKT
TYPE ,CPHALT
HALT
MOV 0#RP4VEC,0#RPVEC

:SAVE REGISTERS
:RHWCSAVERE IS THE FIRST REGISTER SAVED
:STARTING ADDRESS OF WHERE
:THE REGISTERS ARE SAVED
:NUMBER OF REGISTERS
:SAVED = 18.
:CHECKS DVA, RDY MOL, DPR, DRY AND VV = 1
:AND THAT ALL STATUS BITS ARE = 0
:CANNOT CONTINUE TESTING IF NOT
:STOP TEST
:SET RPO4 VECTOR ADDRESS
:TO 'TIME1' IF P-CLOCK IS PRESENT
:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
:'TIME' WILL ONLY SAVE
:CURRENT CYLINDER ADDRESS
:AND LOOK AHEAD REGISTERS

MOV 0#READAT, -(SP)
BIS 0#GO!IE, (SP)

MOV (SP)+,0#RHCSI

:GET READY TO MOVE COMMAND
:GET READY TO SET GO AND
:ENABLE INTERRUPT
:GO WITH
:70 IN RHCSI FOR READ DATA
:WITH INTERRUPT ENABLED

:*TIME IS NOT CRITICAL HERE

WAIT
RHCS1
RDY
908.
834.

:WAIT FOR RDY BIT TO SET
:WAIT FOR RHCS1 REGISTER
:WAIT FOR RDY BIT IN RHCS1 REGISTER
:ALLOW 9080 MICRO SECONDS
:RDY MUST SET BETWEEN
:740 AND 17420 MICRO SECONDS

:*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

JSR RO,0#FILLRE

:MOV 0 INTO SAVED RHWCSAVERE

```

012322 002172 RHWC ;SAVED REGISTER TO CHANGE
012324 000000 0 ;DATA

012326 004037 032706 JSR RO,3#FILLRE ;MOV REINTO+(256.*2) INTO SAVED RHBA
012332 002174 RHBA ;SAVED REGISTER TO CHANGE
012334 004434 REINTO+(256.*2) ;DATA

012336 004037 033674 JSR RO,3#CHREG ;CHANGE BITS IN SAVED REGISTER
012342 002222 RHDS1 ;CHANGE RHDS1 REGISTER

012344 000001 ;1 BIT/BITS TO BE CHANGED
012346 000001 ;NEW VALUE OF LBT IS 1
012350 002000 LBT ;CHANGE LBT BIT

012352 004037 033674 JSR RO,3#CHREG ;CHANGE BITS IN SAVED REGISTER
012356 002212 RHCA ;CHANGE RHCA REGISTER

012360 000001 ;1 BIT/BITS TO BE CHANGED
012362 000001 ;NEW VALUE OF BITD IS 1
012364 000001 BITD ;CHANGE BITD BIT

012366 004037 032706 JSR RO,3#FILLRE ;MOV 0 INTO SAVED RHDST
012372 002204 RHDST ;SAVED REGISTER TO CHANGE
012374 000000 0 ;DATA

; *COMPARE ALL REGISTERS

012376 004037 034002 JSR RO,3#COMREG ;COMPARE SAVED REGISTERS WITH
;PRESENT VALUE
012402 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
012404 002254 NO ;TEST DATA STARTING FROM 'RHWC'
012406 000022 19. ;19. REGISTERS TO BE COMPARED
012410 012414 39. ;RETURN TO 39 ON ERROR
012412 012420 49. ;RETURN TO 49 ON NO ERROR

012414 104045 39: ERROR 45 ;READING ON LAST BLOCK IE.
012416 000207 RTS PC ;CYLINDER 410./814.. SECTOR 21, TRACK 19
;CAUSED AN ERROR
;GOOD DATA GIVES WHAT SHOULD
;BE THERE
;RECEIVED DATA GIVES WHAT
;WAS THERE AFTER READ
;FROM LAST BLOCK

; *READ DATA WILL BE COMPARED

012420 49:

012420 004037 035032 JSR RO,3#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
012424 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
012426 003434 REINTO ;TEST DATA STARTS FROM REINTO

```

E09

MNDFC-11-DZRJJ-A, RFO4 S 6 FUNCT. CONT. TST-PT 2
 DZRJJ.A.P11 †11 LAST BLOCK TRANSFERED-RHDS1 LBT

MACY11 27(655) 30-MAR-76 18:59 PAGE 59

SEQ C107

012430
 012432
 012434
 012436
 012440
 012442
 104046
 000207
 012442

012430 000400
 012432 012436
 012434 012442

256.
 55
 65

:256. WORDS TO BE COMPARED
 :RETURN TO 55 ON ERROR
 :RETURN TO 65 ON NO ERROR

012436 104046
 012440 000207

55: ERROR 46
 RTS PC

:DATA READ FROM
 :LAST BLOCK IN ERROR

012442

65:


```

2628 012530 004737 033004 JSR PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
2629 ;R3-RHDS1, R4-RHER1
2630 ;GIVE RH-11 INITIALIZE
2631 ;SETUP UNIT NUMBER
2632
2633 ;*FILL WRITE FROM BUFFER WITH HEADER
2634 012534 004037 032630 JSR RC,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
2635 012540 002370 WRFROM ;LOCATION WHERE SAVED
2636 012542 000004 4 ;NUMBER OF WORDS SAVED
2637 012544 010000 10000 ;FIRST DATA WORD
2638 012546 000000 0 ;SECOND DATA WORD
2639 012550 000000 0 ;THIRD DATA WORD
2640 012552 000000 0 ;FOURTH DATA WORD
2641
2642 ;*FILL WRITE FROM BUFFER WITH DATA
2643
2644 012554 004037 032654 JSR RC,0#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
2645 012560 002400 WRFROM+10 ;STARTING FROM WRFROM+10
2646 012562 000400 256. ;256. WORDS
2647 012564 000000 0 ;FILL WITH 0
2648
2649 ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
2650
2651 012566 004037 032630 JSR RC,0#FLHEAD ;SAVE HEADER DATA IN WRFROM+(260.*2)
2652 012572 003400 WRFROM+(260.*2) ;LOCATION WHERE SAVED
2653 012574 000004 4 ;NUMBER OF WORDS SAVED
2654 012576 010000 10000 ;FIRST DATA WORD
2655 012600 000001 1 ;SECOND DATA WORD
2656 012602 000000 0 ;THIRD DATA WORD
2657 012604 000000 0 ;FOURTH DATA WORD
2658
2659 ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
2660
2661 012606 004037 032654 JSR RC,0#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+(264.*2)
2662 012612 003410 WRFROM+(264.*2) ;STARTING FROM WRFROM+(264.*2)
2663 012614 000004 4 ;4 WORDS
2664 012616 000001 1 ;FILL WITH 1
2665
2666 ;*CLEAR READ INTO BUFFER WITH DATA OTHER THAN EXPECTED DATA
2667
2668 012620 004037 032654 JSR RC,0#CLAREA ;CLEAR 260. WORDS, FROM REINTO
2669 012624 003434 REINTO ;STARTING FROM REINTO
2670 012626 000404 260. ;260. WORDS
2671 012630 000377 377 ;FILL WITH 377
2672
2673 ;*THE WRITE HEADER AND DATA WILL BE LOADED
2674
2675 012632 004037 034766 JSR RC,0#RUN ;SETUP TO RUN FOR DATA COMMAND
2676 012636 000000 0 ;CYLINDER 0
  
```


2693	012640	000				.BYTE 0	:SECTOR 0
2694	012641	000				.BYTE 0	:TRACK 0
2695	012642	177364				-264.-4	:WORD COUNT (DATA) = 264. +
2696							:4 HEADER WORDS
2697	012644	002370				WRFROM	:BUS ADDRESS
2698							:STARTING ADDRESS OF DATA
2699	012646	000000				0	:BUFFER = WRFROM
2700	012650	010000				FMT22	:DO NOT INHIBIT BUS ADDRESS INCREMENT
2701							:16 BITS PER WORD FORMAT
2702							:DO NOT INHIBIT ECC CORRECTION
2703	012652	002344				WRIFOR	:DO NOT INHIBIT HEADER COMPARE
2704							:GET READY TO DO A WRIFOR
2705							:WRITE HEADER AND DATA WITH 52 IN RHCS1
2706	012654	004037	033152				:*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
2707	012660	002172				JSR RD, @SAVER	:SAVE REGISTERS
2708	012662	004512				RHWC	:RHWC IS THE FIRST REGISTER SAVED
2709						SAVERE	:STARTING ADDRESS OF WHERE
2710							:THE REGISTERS ARE SAVED
2711	012664	000022				18.	:NUMBER OF REGISTERS
2712							:SAVED = 18.
2713							
2714	012666	004737	033064			JSR PC, @CHECKT	:CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
2715							:AND THAT ALL STATUS BITS ARE = 0
2716	012672	104400	056662			TYPE ,CPHALT	:CANNOT CONTINUE TESTING IF NOT
2717	012676	000000				HALT	:STOP TEST
2718							
2719	012700	013777	004506	167260		MOV @RP4VEC, @RPVEC	:SET RPO4 VECTOR ADDRESS
2720							:TO 'TIME1' IF P-CLOCK IS PRESENT
2721							:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
2722							: 'TIME' WILL ONLY SAVE
2723							:CURRENT CYLINDER ADDRESS
2724							:AND LOOK AHEAD REGISTERS
2725							
2726	012706	013746	002344			MOV @WRIFOR, -(SP)	:GET READY TO MOVE COMMAND
2727	012712	052716	000101			BIS @GO!IE, (SP)	:GET READY TO SET GO AND
2728							:ENABLE INTERRUPT
2729	012716	012677	167256			MOV (SP)+, @RHCS1	:GO WITH
2730							:52 IN RHCS1 FOR WRITE HEADER AND DATA
2731							:WITH INTERRUPT ENABLED
2732	012722	011100				MOV @R1, R0	:SAVE RHCS1 DURING ABOVE OPERATION
2733	012724	011305				MOV @R3, R5	:SAVE RHCS1 DURING ABOVE OPERATION
2734							
2735							:*ONE REVOLUTION=16670 MICRO SEC. ONE SECTOR = 760 MICRO SEC
2736	012726	104412				WAT	:WAIT FOR RDY BIT TO SET
2737	012730	002200				RHCS1	:WAIT FOR RHCS1 REGISTER
2738	012732	000200				RDY	:WAIT FOR RDY BIT IN RHCS1 REGISTER
2739	012734	001614				908.	:ALLOW 9080 MICRO SECONDS
2740	012736	001507				839.	:RDY MUST SET BETWEEN
2741							:690 AND 17470 MICRO SECONDS

```

2736                                     : *COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
2737                                     : *R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
2738
2739 012740 013746 002344  MOV      Q#WRIFOR, -(SP)  ; SAVE COMMAND
2740 012744 052716 004101  BIS      #IE!DVA!GO, (SP) ; INCLUDE IE!DVA!GO
2741 012750 011637 001124  MOV      (SP), Q#SGDDAT   ; SAVE FOR PRINTOUT
2742 012754 022600          CMP      (SP)+, R0        ; DURING ABOVE OPERATION ONLY IE!DVA!GO
2743                                     ; AND COMMAND SHOULD BE SET
2744 012756 001405          BEQ      67$             ; BRANCH IF GOOD
2745 012760 010037 001126  MOV      R0, Q#SBDDAT     ; BAD DATA
2746 012764 010137 004500  MOV      R1, Q#REGADR     ; FAILING REGISTER RHCS1
2747 012770 104021          ERROR    21            ; DURING ABOVE OPERATION ONLY
2748                                     ; COMMAND AND IE!DVA!GO SHOULD BE SET
2749 012772 012746 010500 67$:  MOV      #MOL!DPR!VV, -(SP) ; SAVE BITS SET DURING OPERATION IN RHDS1
2750 012776 011637 001124  MOV      (SP), Q#SGDDAT   ; SAVE FOR PRINTOUT
2751 013002 022605          CMP      (SP)+, R5        ; DURING ABOVE OPERATION ONLY MOL!DPR!VV
2752                                     ; SHOULD BE SET
2753 013004 001405          BEQ      69$             ; BRANCH IF GOOD
2754 013006 010537 001126  MOV      R5, Q#SBDDAT     ; BAD DATA
2755 013012 010337 004500  MOV      R3, Q#REGADR     ; FAILING REGISTER RHDS1
2756 013016 104063          ERROR    63            ; DURING ABOVE OPERATION ONLY
2757                                     ; MOL!DPR!VV SHOULD BE SET
2758 013020          69$:
2759
2760                                     : *NOW CHANGE SAVE REGISTERS TO EXPECTED VALUES
2761
2762 013020 004037 032706  JSR      R0, Q#FILLRE     ; MOV 0 INTO SAVED RHWC
2763 013024 002172          RHWC                    ; SAVED REGISTER TO CHANGE
2764 013026 000000          0                      ; DATA
2765
2766
2767 013030 004037 032706  JSR      R0, Q#FILLRE     ; MOV WRFROM-(268.*2) INTO SAVED RHBA
2768 013034 002174          RHBA                    ; SAVED REGISTER TO CHANGE
2769 013036 003420          WRFROM+(268.*2)        ; DATA
2770
2771
2772 013040 004037 032706  JSR      R0, Q#FILLRE     ; MOV 2 INTO SAVED RHDST
2773 013044 002204          RHDST                    ; SAVED REGISTER TO CHANGE
2774 013046 000002          2                      ; DATA
2775
2776
2777                                     : *NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
2778                                     : *WITH REGISTERS AFTER COMMAND
2779
2780 013050 004037 034002  JSR      R0, Q#COMREG     ; COMPARE SAVED REGISTERS WITH
2781                                     ; PRESENT VALUE
2782 013054 004512          SAVERE                    ; GOOD DATA SAVED IN 'SAVERE'
2783 013056 002254          WC                      ; TEST DATA STARTING FROM 'RHWC'
2784 013060 000022          18.                     ; 19. REGISTERS TO BE COMPARED
2785 013062 013066          1$                      ; RETURN TO 1$ ON ERROR
2786 013064 013072          2$                      ; RETURN TO 2$ ON NO ERROR
2787
2788
2789 013066 104027 1$:  ERROR    27            ; WRITE HEADER AND DATA
    
```

```

2790 013070 000207          R/S      PC          ; CAUSED IMPROPER REGISTER
2791                          ; CHANGE
2792                          ; GOOD DATA GIVES WHAT SHOULD
2793                          ; BE THERE
2794                          ; RECEIVED DATA GIVES WHAT
2795                          ; WAS THERE AFTER COMMAND
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808 013072          28:
2809
2810 013072 004737 033004    JSR      PC,0#CLDISK  ; SET R1-RHCS1, R2-RHCS2
2811                          ; R3-RHOS1, R4-RHER1
2812                          ; GIVE RH-11 INITIALIZE
2813                          ; SETUP UNIT NUMBER
2814
2815 013076 004037 034766    JSR      RD,0#RUN     ; SETUP TO RUN FOR DATA COMMAND
2816 013102 000000          0          ; CYLINDER 0
2817 013104          000      .BYTE 0      ; SECTOR 0
2818 013105          000      .BYTE 0      ; TRACK 0
2819 013106 177770          -8.       ; WORD COUNT = 8.
2820 013110 003434          REINTO    ; BUS ADDRESS
2821                          ; STARTING ADDRESS OF DATA
2822                          ; BUFFER = REINTO
2823 013112 000000          0          ; DO NOT INHIBIT BUS ADDRESS INCREMENT
2824 013114 014000          ECI!FMT22 ; 15 BITS PER WORD FORMAT
2825                          ; INHIBIT ECC CORRECTION
2826                          ; DO NOT INHIBIT HEADER COMPARE
2827 013116 002334          SERCH      ; GET READY TO DO A SERCH
2828                          ; SEARCH WITH 30 IN RHCS1
2829
2830
2831
2832
2833 013120 004037 033152    JSR      RD,0#SAVER   ; SAVE REGISTERS
2834 013124 002172          RHC      ; RHC IS THE FIRST REGISTER SAVED
2835 013126 004512          SAVERE    ; STARTING ADDRES OF WHERE
2836                          ; THE REGISTERS ARE SAVED
2837 013130 000022          18.       ; NUMBER OF REGISTERS
2838                          ; SAVED = 18.
2839
2840
2841
2842
2843 013132 013746 002350    MOV      0#REFOR,-(SP) ; SAVE READ HEADER AND DATA
  
```

```

2844 013136 052716 000101      BIS      #IE!GO,(SP)      ;INTERRUPT ENABLE AND GO
2845 013142 012637 004650      MOV      (SP)+, @#TMP0 ;SAVE IN RD FOR RHCS1
2846 013146 012737 000001 004656  MOV      #1, @#TMP5    ;SAVE TRACK 0 SECTOR 1 FOR RHDST
2847
2848 ;*THE INTERRUPT VECTOR WILL BE SET TO GO TO 2$
2849 ;*AFTER THE SEARCH
2850
2851 013154 012777 013222 167004  MOV      #7$, @RPVEC   ;SET INTERRUPT VECTOR TO 2$
2852 013162 004737 033064      JSR      PC, @#CHECKT  ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
2853 ;AND THAT ALL STATUS BITS ARE = 0
2854 013166 104400 056662      TYPE    ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
2855 013172 000000      HALT                    ;STOP TEST
2856
2857 013174 013746 002334      MOV      @#SERCH, -(SP) ;GET READY TO MOVE COMMAND
2858 013200 052716 000101      BIS      #GO!IE,(SP)  ;GET READY TO SET GO AND
2859 ;ENABLE INTERRUPT
2860 013204 012677 166770      MOV      (SP)+, @RHCS1 ;GO WITH
2861 ;WITH INTERRUPT ENABLED
2862
2863 ;*TIME IS NOT CRITICAL THIS ONLY WAITS FOR SEARCH COMPLETION
2864
2865 013210 104412      WAT                    ;WAIT FOR DRY BIT TO SET
2866 013212 002222      RHDS1                 ;WAIT FOR RHDS1 REGISTER
2867 013214 000200      DRY                   ;WAIT FOR DRY BIT IN RHDS1 REGISTER
2868 013216 001614      908.                  ;ALLOW 9080 MICRO SECONDS
2869 013220 001507      839.                  ;DRY MUST SET BETWEEN
2870 ;690 AND 17470 MICRO SECONDS
2871
2872 013222 012737 000000 177776 7$:  MOV      #0,PS         ;SET PROSESSOR STATUS TO
2873 ;PRIORITY 0 IN CASE IT WAS
2874 ;TAKEN OUT OF WAT ROUTINE
2875 ;BEFORE RTI
2876 013230 013777 004656 166746  MOV      @#TMP5, @RHDS1 ;SET DESIRED SECTOR/TRACK
2877 ;REGISTER TO SECTOR 1, TRACK C
2878 013236 013777 004506 166722  MOV      @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
2879 ;TO 'TIME1' IF P-CLOCK IS PRESENT
2880 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
2881 ;'TIME' WILL ONLY SAVE
2882 ;CURRENT CYLINDER ADDRESS
2883 ;AND LOOK AHEAD REGISTERS
2884
2885 013244 013777 004650 166726  MOV      @#TMP0, @RHCS1 ;FILL RHCS1 WITH READ COMMAND
2886 ;TOGETHER WITH INTERRUPT ENABLE
2887 ;AND GO
2888
2889 ;*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
2890 ;*INDICATES WRONG SEARCH IN THE SEARCH COMMAND
2891
2892 013252 104412      WAT                    ;WAIT FOR RDY BIT TO SET
2893 013254 002200      RHCS1                 ;WAIT FOR RHCS1 REGISTER
2894 013256 000200      RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
2895 013260 000225      149.                  ;ALLOW 1490 MICRO SECONDS
2896 013262 000002      2.                   ;RDY MUST SET BETWEEN
2897 ;1470 AND 1510 MICRO SECONDS

```

```

2998
2999
3000
3001 013264 004037 032630 JSR RO,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
3002 013270 002370 WRFROM ;LOCATION WHERE SAVED
3003 013272 000004 4 ;NUMBER OF WORDS SAVED
3004 013274 010000 10000 ;FIRST DATA WORD
3005 013276 000001 1 ;SECOND DATA WORD
3006 013300 000000 0 ;THIRD DATA WORD
3007 013302 000000 0 ;FOURTH DATA WORD
3008
3009 013304 004037 032654 JSR RO,2#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+(4*2)
3010 013310 002400 WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
3011 013312 000004 4 ;4 WORDS
3012 013314 000001 1 ;FILL WITH 1
3013
3014
3015
3016
3017 013316 004037 032706 JSR RO,2#FILLRE ;MOV 0 INTO SAVED RHWC
3018 013322 002172 RHWC ;SAVED REGISTER TO CHANGE
3019 013324 000000 0 ;DATA
3020
3021
3022 013326 004037 032706 JSR RO,2#FILLRE ;MOV REINTO+(8.*2) INTO SAVED RHBA
3023 013332 002174 RHBA ;SAVED REGISTER TO CHANGE
3024 013334 003454 REINTO+(8.*2) ;DATA
3025
3026
3027 013336 004037 032706 JSR RO,2#FILLRE ;MOV 4272 INTO SAVED RHCS1
3028 013342 002200 RHCS1 ;SAVED REGISTER TO CHANGE
3029 013344 004272 4272 ;DATA
3030
3031
3032 013346 004037 032706 JSR RO,2#FILLRE ;MOV 2 INTO SAVED RHDST
3033 013352 002204 RHDST ;SAVED REGISTER TO CHANGE
3034 013354 000002 2 ;DATA
3035
3036
3037
3038
3039
3040
3041 013356 004037 034002 JSR RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
3042 ;PRESENT VALUE
3043 013362 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3044 013364 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3045 013366 000022 18. ;18. REGISTERS TO BE COMPARED
3046 013370 013374 3$ ;RETURN TO 3$ ON ERROR
3047 013372 013400 4$ ;RETURN TO 4$ ON NO ERROR
3048
3049 013374 104031 3$: ERRC? 31 ;READ HEADER AND DATA CAUSED
3050 013376 000207 RTS PC ;IMPROPER REGISTER CHANGE
3051 ;GOOD DATA GIVES WHAT SHOULD

```

2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972

013400

4\$:

013400 004037 035032

013404 002370

013406 003434

013410 000010

013412 013416

013414 013422

013416 104053

013420 000207

013422

5\$:

6\$:

JSR RD,2#COMPAR
WRFROM
REINTO
8.
5\$
6\$

:BE THERE
:RECEIVED DATA GIVES WHAT WAS
:THERE AFTER COMMAND

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

:COMPARE TWO BLOCKS OF MEMORY
:GOOD DATA STARTS FROM WRFROM
:TEST DATA STARTS FROM REINTO
:8. WORDS TO BE COMPARED
:RETURN TO 5\$ ON ERROR
:RETURN TO 6\$ ON NO ERROR

:READ HEADER AND DATA
:AFTER A SEARCH CAUSED
:AN ERROR

2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026

: *TEST 13 SEARCH COMMAND

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

TST13: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #13, @#TSTNM ;SAVE TEST NUMBER
JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
; *GET THE HEADS TO CYLINDER 10
JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST
MOV @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS
JSR RD, @#SEEKCY ;SEEK FOR
10. ;CYLINDER 10.
MOV @#SEECOM, -(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT
MOV (SP)+, @RHCS1 ;GO WITH
;4 IN RHCS1 FOR SEEK
;WITH INTERRUPT ENABLED
WAT ;WAIT FOR DRY BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7000. ;ALLOW 70000 MICRO SECONDS
35. ;DRY MUST SET BETWEEN
;69650 AND 70350 MICRO SECONDS

;*FILL REGISTERS FOR READ HEADER AND DATA TO BE DONE AFTER SEARCH

013516 004737 033004
013522 004037 034766
013523 000000
013524 000000
013525 000000
013526 000000
013527 177777
013528 003434
013536 000000
013540 014000
013542 002334
013544 004037 033152
013550 002172
013552 004512
013554 000022
013556 004737 033064
013562 104400 056662
013566 000000
013570 012777 013754 166370
013576 004037 032734
013602 000000
013604 000
013605 000
013606 .BYTE 0
013607 .BYTE 0
013608 .BYTE 0
013609 .BYTE 0
013610 .BYTE 0
013611 .BYTE 0
013612 052700 004301
013616 012705 010500
013622 013746 002334
013626 052716 000101

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

JSR RD, @RUN ;SETUP TO RUN FOR DATA COMMAND
C ;CYLINDER 0
;SECTOR 0
.BYTE 0 ;TRACK 0
.BYTE 0 ;WORD COUNT = 8.
-8 ;BUS ADDRESS
REINTO ;STARTING ADDRESS OF DATA
;BUFFER = REINTO

0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECC:FM*22 ;16 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
SERCH ;GET READY TO DO A SERCH
;SEARCH WITH 30 IN RHCS1

;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH

;*AND READ HEADER AND DATA
JSR RD, @SAVER ;SAVE REGISTERS
RHW ;RHW IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
18. ;NUMBER OF REGISTERS
;SAVED = 18.

;*NOW GIVE THE SEARCH COMMAND

JSR PC, @CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST
MOV #3\$, @RPVEC ;INTERRUPT VECTOR SET TO 3\$

JSR RD, @SRCH ;SEARCH FOR
0 ;CYLINDER 0
0 ;SECTOR 0
0 ;TRACK 0

MOV @SERCH, RD ;EXPECTED CONTENTS OF RHCS1
;IMMEDIATELY AFTER GO
BIS #DVA!RDY!IE!GO, RD ;EXPECTED BITS IN RHCS1
MOV #MOL!DPR!VV, RS ;EXPECTED BITS IN RHDS1
;IMMEDIATELY AFTER GO

MOV @SERCH, -(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT


```

013632 012677 166342      MOV      (SP)+, @RHCS1      :GO WITH
                                :WITH INTERRUPT ENABLED
013636 021100      CMP      @R1, @R0          :IS RHCS1 GOOD
                                :BRANCH IF GOOD
013637 001413      BEQ     15                 :BAD DATA FOR RHCS1
                                :GOOD DATA
013638 011133      MOV     @R1, @S$BDDAT      :FADING REGISTER RHCS1
                                :TRAP PRIORITY = 7
013639 010033      MOV     @R0, @S$GDDAT      :DURING SEARCH COMMAND
                                :CONTENTS OF RHCS1 WAS
013640 004500      MOV     @R1, @RREGADR      :NOT AS EXPECTED
                                :IF LAST ERROR 21 OCCURRED
013641 000240      MOV     @340, @TRAPVEC+2  :THEN DO NOT CHECK RHDS1
                                :AS TOO MUCH TIME HAS
                                :PASSED
                                :
013656 000414      BR      25
                                :
15: 013657 021133      CMP     @R3, @R5           :IS RHDS1 GOOD
                                :BRANCH IF GOOD
013658 001125      BEQ     25                 :BAD DATA FOR RHDS1
                                :GOOD DATA
013659 010333      MOV     @R3, @S$BDDAT      :FADING REGISTER RHDS1
                                :TRAP PRIORITY = 7
013660 004500      MOV     @R3, @RREGADR      :DURING SEARCH COMMAND
                                :CONTENTS OF RHDS1 WAS
013661 000340      MOV     @340, @TRAPVEC+2  :IN CORRECT
                                :
25: 013720 012737 002350 004650      MOV     @REFOR, @TMPD      :SAVE READ HEADER AND DATA
013721 000101 004650      BIS     @TE!GO, @TMPD     :INCLUDE INTERRUPT ENABLE. GO
013722 012737 000001 004656      MOV     @1, @TMP5         :SAVE TRACK 0, SECTOR 1
                                :
                                :*THIS IS ONLY A WAIT LOOP
                                :
013724 104410      JAT     @RDY              :WAIT FOR RDY BIT TO SET
013725 000200      RHDS1                    :WAIT FOR RHDS1 REGISTER
013726 000000      RDY                      :WAIT FOR RDY BIT IN RHDS1 REGISTER
013727 010000      7000.                   :ALLOW 7000 MICRO SECONDS
013728 000043      35.                     :RDY MUST SET BETWEEN
                                :69650 AND 70350 MICRO SECONDS
                                :
013734 012737 000200 000036 35:  MOV     @200, @TRAPVEC+2  :TRAP PRIORITY = 4
013735 000000 177776      MOV     @0, @PS           :SET PROSESSOR STATUS TO 0
013736 013777 004656 166206      MOV     @TMP5, @RHDS1     :SET DESIRED SECTOR/TRACK
                                :REGISTER TO SECTOR 1, TRACK 0
                                :
013776 013777 004506 166162      MOV     @RPHVEC, @RPVEC   :SET RPO4 VECTOR ADDRESS
                                :TO 'TIME1' IF P-CLOCK IS PRESENT
                                :OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                :'TIME' WILL ONLY SAVE
                                :CURRENT CYLINDER ADDRESS
                                :AND LOOK AHEAD REGISTERS
014004 013711 004650      MOV     @TMPD, @R1        :FILL RHCS1 WITH READ COMMAND
                                :TOGETHER WITH INTERRUPT ENABLE
                                :AND GO

```

014010
014012
014014
014016
014020
014022
014026
014030
014032
014034
014036
014040
014042
014046
014050
014052
014054
014060
014062
014064
014070
014072
014074
014100
014102
014104
014110
014112
014114
014120
014122

104412
002230
000200
000225
000002

004037 032630
002370
000004
010000
000001
000000
000000

004037 032654
002400
000004
000001

004037 032706
002172
000000

004037 032706
002174
003454

004037 032706
002300
004272

004037 032706
002204
000002

004037 032706
002234
000000

:*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
:*INDICATES WRONG SEARCH IN THE SEARCH COMMAND

WAT :WAIT FOR RDY BIT TO SET
RHCS1 :WAIT FOR RHCS1 REGISTER
RDY :WAIT FOR RDY BIT IN RHCS1 REGISTER
149. :ALLOW 1490 MICRO SECONDS
2. :RDY MUST SET BETWEEN
:1470 AND 1510 MICRO SECONDS

:*WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA

JSR RD,0#FLHEAD :SAVE HEADER DATA IN WRFROM
WRFROM :LOCATION WHERE SAVED
4 :NUMBER OF WORDS SAVED
10000 :FIRST DATA WORD
1 :SECOND DATA WORD
0 :THIRD DATA WORD
0 :FOURTH DATA WORD

JSR RD,0#CLAREA :CLEAR 4 WORDS FROM WRFROM+(4*2)
WRFROM+(4*2) :STARTING FROM WRFROM+(4*2)
4 :4 WORDS
1 :FILL WITH 1

:*CHANGE SAVED REGISTERS TO EXPECTED VALUES

JSR RD,0#FILLRE :MOV 0 INTO SAVED RHWC
RHWC :SAVED REGISTER TO CHANGE
0 :DATA

JSR RD,0#FILLRE :MOV REINT0+(8.*2) INTO SAVED RHBA
RHBA :SAVED REGISTER TO CHANGE
REINT0+(8.*2) :DATA

JSR RD,0#FILLRE :MOV 4272 INTO SAVED RHCS1
RHCS1 :SAVED REGISTER TO CHANGE
4272 :DATA

JSR RD,0#FILLRE :MOV 2 INTO SAVED RHDST
RHCS1 :SAVED REGISTER TO CHANGE
2 :DATA

JSR RD,0#FILLRE :MOV 0 INTO SAVED RHCC
RHCC :SAVED REGISTER TO CHANGE
0 :DATA

:*COMPARE REGISTER BEFORE READ HEADER AND DATA

:*WITH REGISTERS AFTER COMMAND

3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500

014124 004037 034002
014130 004512
014132 002254
014134 000022
014136 014143
014140 014146

JSR RO,2#COMREG
SAVERE
WC
18.
4\$
5\$

:COMPARE SAVED REGISTERS WITH
:PRESENT VALUE
:GOOD DATA SAVED IN 'SAVERE'
:TEST DATA STARTING FROM 'RHWC'
:18. REGISTERS TO BE COMPARED
:RETURN TO 4\$ ON ERROR
:RETURN TO 5\$ ON NO ERROR

014142 104031
014144 000207

4\$: ERROR 31
RTS PC

:READ HEADER AND DATA CAUSED
:IMPROPER REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD
:BE THERE
:RECEIVED DATA GIVES WHAT WAS
:THERE AFTER COMMAND

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

014146
014146 004037 035032
014150 002370
014154 003434
014156 000010
014160 014164
014162 014170

JSR RO,2#COMPAR
WRFROM
REINTO
8.
6\$
7\$

:COMPARE TWO BLOCKS OF MEMORY
:GOOD DATA STARTS FROM WRFROM
:TEST DATA STARTS FROM REINTO
:8. WORDS TO BE COMPARED
:RETURN TO 6\$ ON ERROR
:RETURN TO 7\$ ON NO ERROR

014164 104053
014166 000207

6\$: ERROR 53
RTS PC

:READ HEADER AND DATA
:AFTER A SEARCH CAUSED
:AN ERROR

014170

7\$:


```

02293 014232 004037 032654 JSR RO, @#CLAREA ;CLEAR 256. WORDS, FROM REINTO
02294 014236 003434 REINTO ;STARTING FROM REINTO
02295 014240 000400 256. ;256. WORDS
02296 014242 177777 -1 ;FILL WITH -1
02297
02298 ;*WRITE HEADER AND DATA IS LOADED
02299
02300 014244 004037 034766 JSR RO, @#RUN ;SETUP TO RUN FOR DATA COMMAND
02301 014250 000000 0. ;CYLINDER 0.
02302 014252 001 .BYTE 1. ;SECTOR 1.
02303 014253 000 .BYTE 0. ;TRACK 0.
02304 014254 177777 -1-4 ;WORD COUNT (DATA) = 1 +
02305 ;4 HEADER WORDS
02306 014256 002370 WRFROM ;BUS ADDRESS
02307 ;STARTING ADDRESS OF DATA
02308 ;BUFFER = WRFROM
02309 014260 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
02310 014262 010000 FMT22 ;16 BITS PER WORD FORMAT
02311 ;DO NOT INHIBIT ECC CORRECTION
02312 ;DO NOT INHIBIT HEADER COMPARE
02313 014264 002344 WRIFOR ;GET READY TO DO A WRIFOR
02314 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
02315
02316 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE
02317
02318 014266 004037 033152 JSR RO, @#SAVER ;SAVE REGISTERS
02319 014272 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
02320 014274 004512 SAVERE ;STARTING ADDRESS OF WHERE
02321 ;THE REGISTERS ARE SAVED
02322 014276 000021 17. ;NUMBER OF REGISTERS
02323 ;SAVED = 17.
02324
02325 014300 004737 033064 JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
02326 ;AND THAT ALL STATUS BITS ARE = 0
02327 014304 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
02328 014310 000000 HALT ;STOP TEST
02329
02330 014312 013777 004506 165646 MOV @#RPHVEC, @#RPVEC ;SET RPO4 VECTOR ADDRESS
02331 ;TO 'TIME1' IF P-CLOCK IS PRESENT
02332 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
02333 ;'TIME' WILL ONLY SAVE
02334 ;CURRENT CYLINDER ADDRESS
02335 ;AND LOOK AHEAD REGISTERS
02336
02337 014320 013746 002344 MOV @#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
02338 014324 052716 000101 BIS @#GO!IE, (SP) ;GET READY TO SET GO AND
02339 ;ENABLE INTERRUPT
02340 014330 012677 165644 MOV (SP)+, @#RHCS1 ;GO WITH
02341 ;62 IN RHCS1 FOR WRITE HEADER AND DATA

```

;WITH INTERRUPT ENABLED

;*TIME IS NOT CRITICAL

3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381

014334 104412
014336 002200
014340 000200
014342 004704
014344 004704

WAT
RHCS1
RDY
2500.
2500.

;WAIT FOR RDY BIT TO SET
;WAIT FOR RHCS1 REGISTER
;WAIT FOR RDY BIT IN RHCS1 REGISTER
;ALLOW 25000 MICRO SECONDS
;RDY MUST SET BETWEEN
;00 AND 50000 MICRO SECONDS

;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

014346 004037 032706
014352 002172
014354 000000

JSR RO,2#FILLRE
RHWC
0

;MOV 0 INTO SAVED RHWC
;SAVED REGISTER TO CHANGE
;DATA

014356 004037 032706
014362 002174
014364 002402

JSR RO,2#FILLRE
RHBA
WRFROM+5*2>

;MOV WRFROM+5*2> INTO SAVED RHBA
;SAVED REGISTER TO CHANGE
;DATA

014366 004037 032706
014372 002204
014374 000002

JSR RO,2#FILLRE
RHDST
2

;MOV 2 INTO SAVED RHDST
;SAVED REGISTER TO CHANGE
;DATA

;*COMPARE ALL REGISTERS

014376 004037 034002
014402 004512
014404 002254
014406 000021
014410 014414
014412 014420

JSR RO,2#COMREG
SAVERE
WC
17.
1\$
2\$

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

014414 104027 13:
014416 000207

ERROR 27
RTS PC

;WRITING HEADER AND DATA CALSED
;IMPROPER REGISTER CHANGE
;GOOD DATA GIVES WHAT
;SHOULD BE THERE
;RECEIVED DATA GIVES WHAT
;WAS THERE AFTER WRITE

03392
03393
03394
03395
03396
03397
03398
03399
03400
03401
03402
03403
03404
03405
03406
03407
03408
03409
03410
03411
03412
03413
03414
03415
03416
03417
03418
03419
03420
03421
03422
03423
03424
03425

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

014420 25:

014420 004737 033004 JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

014424 004037 032654 JSR RD, @#CLAREA ;CLEAR 256. WORDS FROM WRFROM
014430 002370 WRFROM ;STARTING FROM WRFROM
014432 000400 256. ;256. WORDS
014434 177777 -1 ;FILL WITH -1

;*READ COMMAND IS LOADED

014436 004037 034766 JSR RD, @#RUN ;SETUP TO RUN FOR DATA COMMAND
014442 000000 0 ;CYLINDER 0
014444 001 .BYTE 1. ;SECTOR 1.
014445 000 .BYTE 0 ;TRACK 0
014446 177777 -1 ;WORD COUNT = 1
014450 003434 REINTC ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTC
014452 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
014454 010000 FMT22 ;16 BITS PER WORD FORMAT
;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
014456 002346 READAT ;GET READY TO DO A READAT
;READ DATA WITH 70 IN RHCS1

;*SAVE REGISTERS FOR COMPARISON AFTER READ DATA

014460 004037 033152 JSR RD, @#SAVER ;SAVE REGISTERS
014464 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
014466 004512 SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
014470 000022 18. ;NUMBER OF REGISTERS
;SAVED = 18.

014472 004737 033064 JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
014476 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
014502 000000 HALT ;STOP TEST

014504 013777 004506 165454 MOV @#RP4VEC, @#RPVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT HERE
;'TIME' WILL ONLY SAVE

K10

```

3490 ;SAVED RHAS LOCATION
3491
3492 ;*COMPARE ALL REGISTERS
3493
3494 014632 004037 024002 JSR RO,3#COMREG ;COMPARE SAVED REGISTERS WITH
3495 ;PRESENT VALUE
3496 014636 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3497 014640 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3498 014642 000022 18. ;18. REGISTERS TO BE COMPARED
3499 014644 014650 3$ ;RETURN TO 3$ ON ERROR
3500 014646 014654 4$ ;RETURN TO 4$ ON NO ERROR
3501
3502 014650 104047 3$: ERROR 47 ;READING ON NON EXISTANT SECTOR
3503 014652 000207 RTS PC
3504 ;CAUSED AN ERROR
3505 ;GOOD DATA GIVES WHAT SHOULD
3506 ;BE THERE
3507 ;RECEIVED DATA GIVES WHAT
3508 ;WAS THERE AFTER READ
3509
3510 ;*READ DATA WILL BE COMPARED
3511
3512 014654 4$:
3513
3514 014654 004037 035032 JSR RO,3#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
3515 014660 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
3516 014662 003434 REINTO ;TEST DATA STARTS FROM REINTO
3517 014664 000400 256. ;256. WORDS TO BE COMPARED
3518 014666 014672 5$ ;RETURN TO 5$ ON ERROR
3519 014670 014676 6$ ;RETURN TO 6$ ON NO ERROR
3520
3521 014672 104050 5$: ERROR 50 ;DATA READ FROM NON
3522 014674 000207 RTS PC ;EXISTANT SECTOR CAUSED AN ERROR
3523 014676 6$:
3524

```

3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578

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

;* WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
;* AND PUT SECTOR 0 ON CYLINDER 0
;* THEN A WRITE DATA IS GIVEN FOR SECTOR 1, TRACK 0, CYLINDER 0
;* FOR 70. WORDS
;* HCE - BIT 7 IN RHER1 SHOULD SET
;* ALL REGISTERS ARE CHECKED
;* THEN A READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
;* IS GIVEN, HCE - BIT 7 SHOULD SET AND ALL
;* HEADER AND DATA SHOULD BE READ

::*****

014676 000004
014700 012706 001000
014704 012737 000015 004504

014712 004737 033004

014716 004037 032630
014722 002370
014724 000006
014726 010000
014730 000000
014732 000000
014734 000000
014736 000001
014740 000001

TST15: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #15,2#TSTNM ;SAVE TEST NUMBER

JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

;*FILL WRITE FROM BUFFER WITH HEADER AND DATA

JSR RO,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
6 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
1 ;FIFTH DATA WORD
1 ;SIXTH DATA WORD

;*FILL READ INTO BUFFER WITH ALL ONES

JSR RO,2#CLAREA ;CLEAR 256. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
256. ;256. WORDS
-1 ;FILL WITH -1

;*WRITE HEADER AND DATA IS LOADED

JSR RO,2#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 1 ;SECTOR 1
.BYTE 0 ;TRACK 0
-2-4 ;WORD COUNT (DATA) = 2 +
;4 HEADER WORDS
WRFROM ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM

M10

```

3579 014770 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3580 014772 010000 FMT22 ;16 BITS PER WORD FORMAT
3581 ;DO NOT INHIBIT ECC CORRECTION
3582 ;DO NOT INHIBIT HEADER COMPARE
3583 014774 002344 WRIFOR ;GET READY TO DO A WRIFOR
3584 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
3585
3586
3587 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER
3588 ;*WRITE HEADER AND DATA
3589
3590 014776 004037 033152 JSR RD, @#SAVER ;SAVE REGISTERS
3591 015002 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
3592 015004 004512 SAVERE ;STARTING ADDRESS OF WHERE
3593 ;THE REGISTERS ARE SAVED
3594 015006 000021 17. ;NUMBER OF REGISTERS
3595 ;SAVED = 17.
3596
3597 015010 004737 033064 JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
3598 ;AND THAT ALL STATUS BITS ARE = 0
3599 015014 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
3600 015020 000000 HALT ;STOP TEST
3601
3602 015022 013777 004506 165136 MOV @#RPHVEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
3603 ;TO 'TIME1' IF P-CLOCK IS PRESENT
3604 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3605 ;'TIME' WILL ONLY SAVE
3606 ;CURRENT CYLINDER ADDRESS
3607 ;AND LOOK AHEAD REGISTERS
3608
3609
3610 015030 013746 002344 MOV @#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
3611 015034 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
3612 ;ENABLE INTERRUPT
3613 015040 012677 165134 MOV (SP)+, @RHCS1 ;GO WITH
3614 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
3615 ;WITH INTERRUPT ENABLED
3616
3617 ;*TIME IS NOT CRITICAL
3618
3619 015044 104412 WAT ;WAIT FOR RDY BIT TO SET
3620 015046 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
3621 015050 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3622 015052 004704 2500. ;ALLOW 25000 MICRO SECONDS
3623 015054 004704 2500. ;RDY MUST SET BETWEEN
3624 ;00 AND 50000 MICRO SECONDS
3625
3626 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3627
3628 015056 004037 032706 JSR RD, @#FILLRE ;MOV 0 INTO SAVED RHWC
3629 015062 002172 RHWC ;SAVED REGISTER TO CHANGE
3630 015064 000000 0 ;DATA
3631
3632
    
```

```

3633 015066 004037 032706 JSR RO,2#FILLRE ;MOV WRFROM+<6*2> INTO SAVED RHBA
3634 015072 002174 RHBA ;SAVED REGISTER TO CHANGE
3635 015074 002404 WRFROM+<6*2> ;DATA
3636
3637
3638 015076 004037 032706 JSR RO,2#FILLRE ;MOV 2 INTO SAVED RHDST
3639 015102 002204 RHDST ;SAVED REGISTER TO CHANGE
3640 015104 000002 2 ;DATA
3641
3642
3643 ;*COMPARE ALL REGISTERS
3644
3645 015106 004037 034002 JSR RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
3646 ;PRESENT VALUE
3647 015112 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3648 015114 002254 WC ;TEST DATA STARTING FROM 'RHWC'
3649 015116 000021 17. ;17. REGISTERS TO BE COMPARED
3650 015120 015124 1$ ;RETURN TO 1$ ON EFROR
3651 015122 015130 2$ ;RETURN TO 2$ ON NO ERROR
3652
3653 015124 104027 1$: ERROR 27 ;WRITING HEADER AND DATA CAUSED
3654 015126 000207 RTS PC ;IMPROPER REGISTER CHANGE
3655 ;GOOD DATA GIVES WHAT
3656 ;SHOULD BE THERE
3657 ;RECEIVED DATA GIVES WHAT
3658 ;WAS THERE AFTER WRITE
3659

```

3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713

015130 28:
015136
015156
015162
015164 004037 033152
015170 002172
015172 004512
015174 000022
015176 004737 03330f
015202 104400 056602
015206 000000
015210 013777 004506 164750
015216 013746 002342
015222 002116 000101

: *NOW A WRITE DATA WILL BE DONE ON SAME CYLINDER, SECTOR
: *TRACK

: *FILL WRITE FROM BUFFER WITH DATA

JSR RO, @CLAREA : CLEAR 70. WORDS, FROM WRFROM
WRFROM : STARTING FROM WRFROM
70 : 70. WORDS
177400 : FILL WITH 177400

: *WRITE DATA COMMAND IS LOADED

JSR RO, @RUN : SETUP TO RUN FOR DATA COMMAND
CYLINDER 0 : CYLINDER 0
SECTOR 1 : SECTOR 1
TRACK 0 : TRACK 0
WORD COUNT = 70. : WORD COUNT = 70.
BUS ADDRESS : BUS ADDRESS
STARTING ADDRESS OF DATA : STARTING ADDRESS OF DATA
BUFFER = WRFROM : BUFFER = WRFROM

0 : DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 : 16 BITS PER WORD FORMAT
WRIDAT : DO NOT INHIBIT ECC CORRECTION
GET READY TO DO A WRIDAT : DO NOT INHIBIT HEADER COMPARE
: WRITE DATA WITH 60 IN RHCS1

: *SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA

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

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

MOV @RPN4VEC, @RPVEC : SET RPN4 VECTOR ADDRESS
TO 'TIME1' IF P-CLOCK IS PRESENT
OR TO 'TIME2' IF P-CLOCK IS NOT THERE
'TIME' WILL ONLY SAVE : 'TIME' WILL ONLY SAVE
CURRENT CYLINDER ADDRESS : CURRENT CYLINDER ADDRESS
AND LOOK AHEAD REGISTERS : AND LOOK AHEAD REGISTERS

MOV @WRIDAT, -(SP) : GET READY TO MOVE COMMAND
BIS @GO!IE, (SP) : GET READY TO SET GO AND

015226
015227
015228
015229
015230
015231
015232
015233
015234
015235
015236
015237
015238
015239
015240
015241
015242
015243
015244
015245
015246
015247
015248
015249
015250
015251
015252
015253
015254
015255
015256
015257
015258
015259
015260
015261
015262
015263
015264
015265
015266
015267
015268
015269
015270
015271
015272
015273
015274
015275
015276
015277
015278
015279
015280
015281
015282
015283
015284
015285
015286
015287
015288
015289
015290
015291
015292
015293
015294
015295
015296
015297
015298
015299
015300
015301
015302
015303
015304
015305
015306
015307
015308
015309
015310
015311
015312
015313
015314
015315
015316
015317
015318
015319
015320
015321
015322
015323
015324
015325
015326
015327
015328
015329
015330
015331
015332
015333
015334
015335
015336
015337
015338
015339
015340

```

015226 012677 164746      MOV      (SP)+,0RHCS1      ;ENABLE INTERRUPT
                                           ;GO WITH
                                           ;60 IN RHCS1 FOR WRITE DATA
                                           ;WITH INTERRUPT ENABLED

                                           ;*TIME IS NOT CRITICAL

015233 104412      WAT
015234 002200      RHCS1                      ;WAIT FOR RDY BIT TO SET
015235 000200      RDY                        ;WAIT FOR RHCS1 REGISTER
015236 001614      908.                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
015237 001502      934.                       ;ALLOW 9080 MICRO SECONDS
                                           ;RDY MUST SET BETWEEN
                                           ;740 AND 17420 MICRO SECONDS

015244 005737 004640      ;*NOW CHANGE REGISTERS TO EXPECTED VALUE
015245 001421      TST      0#RH70            ;RUNNING ON RH70 ?
                                           BEQ      95                ;IF NOT, SKIP NEXT RH70 CODE

015253 004037 032706      JSR      RD,0#FILLRE       ;MOV -62. INTO SAVED RHWC
015254 002172      RHWC                       ;SAVED REGISTER TO CHANGE
015255 177702      -62.                       ;DATA

015263 004037 032706      JSR      RD,0#FILLRE       ;MOV WRFROM+(8.*2) INTO SAVED RHBA
015264 002174      RHBA                       ;SAVED REGISTER TO CHANGE
015265 002410      WRFROM+(8.*2)             ;DATA

015272 004037 033674      JSR      RD,0#CHREG        ;CHANGE BITS IN SAVED REGISTER
015273 002176      RHCS2                      ;CHANGE RHCS2 REGISTER

015300 000002      2                          ;2 BIT/BITS TO BE CHANGED
015301 000001      1                          ;NEW VALUE OF OR IS 1
015302 000200      OR                          ;CHANGE OR BIT
015303 000200      0                          ;NEW VALUE OF IR IS 0
015304 000100      IR                          ;CHANGE IR BIT
015305 000416      BR      105                ;SKIP NEXT RH11 CODE

015314      95:

015314 004037 032706      JSR      RD,0#FILLRE       ;MOV -4. INTO SAVED RHWC
015315 002172      RHWC                       ;SAVED REGISTER TO CHANGE
015316 177774      -4.                        ;DATA

015324 004037 032706      JSR      RD,0#FILLRE       ;MOV WRFROM+(66.*2) INTO SAVED RHBA
015325 002174      RHBA                       ;SAVED REGISTER TO CHANGE
015326 002574      WRFROM+(66.*2)           ;DATA

015334 004037 033674      JSR      RD,0#CHREG        ;CHANGE BITS IN SAVED REGISTER
015335 002176      RHCS2                      ;CHANGE RHCS2 REGISTER

```


:CYLINDER 0, SECTOR 1, TRACK 0
:GOOD DATA GIVES WHAT SHOULD BE THERE
:RECEIVED DATA GIVES WHAT WAS THERE
:AFTER COMMAND

:*READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
:*WILL BE ATTEMPTED

015464

48:

015464 004737 033004

JSR PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER;
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

:*FILL WRITE FROM BUFFER WITH EXPECTED DATA

015470 004037 032630

JSR RO,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
6 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
0 ;FIFTH DATA WORD
1 ;SIXTH DATA WORD
1

015504 004037 032654

JSR RO,0#CLAREA ;CLEAR 198. WORDS, FROM WRFROM+(6*2)
WRFROM+(6*2) ;STARTING FROM WRFROM+(6*2)
198. ;198. WORDS
0 ;FILL WITH 0

015526 004037 032654

JSR RO,0#CLAREA ;CLEAR 50. WORDS, FROM WRFROM+ 204.*2
WRFROM+(204.*2) ;STARTING FROM WRFROM+(204.*2)
50. ;50. WORDS
-1 ;FILL WITH -1

:*FILL READ INTO BUFFER WITH ALL ONES

015540 004037 032654

JSR RO,0#CLAREA ;CLEAR 256. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
256. ;256. WORDS
-1 ;FILL WITH -1

:*FILL REGISTERS WITH READ HEADER AND DATA COMMAND

015552 004037 034766

JSR RO,0#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 1 ;SECTOR 1
.BYTE 0 ;TRACK 0
-200.-4 ;WORD COUNT (DATA) = 200. +
;4 HEADER WORDS

F11

```

015564 003434 REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO
015566 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
015570 014000 ECI:FMT22 ;16 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
015572 002350 REFOR ;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A REFOR
;READ HEADER AND DATA WITH 72 IN RHCS1

; *SAVE REGISTERS FOR COMPARISON AFTER READ
; *HEADER AND DATA
015574 004037 033152 JSR RO, @SAVER ;SAVE REGISTERS
015600 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
015602 004512 SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
015604 000022 18. ;NUMBER OF REGISTERS
;SAVED = 18.

015605 004737 033064 JSR PC, @CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
015612 104400 056662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
015616 000000 HALT ;STOP TEST

015620 013777 004506 164340 MOV @RPHVEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
; 'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS

015626 013746 002350 MOV @REFOR, -(SP) ;GET READY TO MOVE COMMAND
015632 052716 000101 BIS @GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT
015636 012677 164326 MOV (SP)+, @RHCS1 ;GO WITH
;72 IN RHCS1 FOR READ DATA
;WITH INTERRUPT ENABLED

; *TIME IS NOT CRITICAL

015642 104412 WAIT ;WAIT FOR RDY BIT TO SET
015644 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
015646 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
015650 001614 909. ;ALLOW 9090 MICRO SECONDS
015652 001507 839. ;RDY MUST SET BETWEEN
;690 AND 17470 MICRO SECONDS

; *CHANGE SAVED REGISTERS TO EXPECTED VALUE

015654 004037 032706 JSR RO, @FILLRE ;MOV 0 INTO SAVED RHWC
015660 002172 RHWC ;SAVED REGISTER TO CHANGE
015662 000000 0 ;DATA
  
```

```

03930
03931
03932 015664 004037 032706 JSR RO,2#FILLRE ;MOV REINTO+(204.*2) INTO SAVED RHBA
03933 015670 002174 RH3A ;SAVED REGISTER TO CHANGE
03934 015672 004264 REINTO+(204.*2) ;DATA
03935
03936
03937 015674 004037 033674 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
03938 015700 002202 RHER1 ;CHANGE RHER1 REGISTER
03939
03940 015702 000001 1 ;1 BIT/BITS TO BE CHANGED
03941 015704 000001 1 ;NEW VALUE OF HCE IS 1
03942 015706 000200 HCE ;CHANGE HCE BIT
03943
03944 015710 004037 033674 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
03945 015714 002222 RHDS1 ;CHANGE RHDS1 REGISTER
03946
03947 015716 000002 2 ;2 BIT/BITS TO BE CHANGED
03948 015720 000001 1 ;NEW VALUE OF ATA IS 1
03949 015722 100000 ATA ;CHANGE ATA BIT
03950 015724 000001 1 ;NEW VALUE OF ERR IS 1
03951 015726 040000 ERR ;CHANGE ERR BIT
03952
03953 015730 004037 032706 JSR RO,2#FILLRE ;MOV 2 INTO SAVED RHDST
03954 015734 002204 RHDST ;SAVED REGISTER TO CHANGE
03955 015736 000002 2 ;DATA
03956
03957
03958 015740 004037 033674 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
03959 015744 002200 RHCS1 ;CHANGE RHCS1 REGISTER
03960
03961 015746 000001 1 ;1 BIT/BITS TO BE CHANGED
03962 015750 000001 1 ;NEW VALUE OF SC!TRE IS 1
03963 015752 140000 SC!TRE ;CHANGE SC!TRE BIT
03964
03965 015754 053737 004644 004536 BIS 2#ATTENT,2#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
03966 ;FOR WORKING DRIVE IN
03967 ;SAVED RHAS LOCATION
03968
03969
03970 ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
03971 ;*WITH AFTER
03972 015762 004037 034002 JSR RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
03973 ;PRESENT VALUE
03974 015766 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
03975 015770 002254 WC ;TEST DATA STARTING FROM 'RHWC'
03976 015772 000022 18. ;18. REGISTERS TO BE COMPAREL
03977 015774 016000 5$ ;RETURN TO 5$ ON ERROR
03978 015776 016004 6$ ;RETURN TO 6$ ON NO ERROR
03979
03980
03981 016000 104031 5$: ERROR 31 ;READ HEADER AND DATA WITH
03982 016002 000207 RTS PC ;FORCED HEADER COMPARE ERROR
03983 ;CAUSED ERROR

```

: GOOD DATA GIVES WHAT SHOULD
: BE THERE
: RECEIVED DATA GIVES WHAT
: WAS THERE AFTER READ

: *NOW COMPARE READ DATA
: *THE COMMAND READ ONLY 204 WORDS, 4 HEADER WORDS
: *AND 200 DATA WORDS

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

016004

6\$:

016004 304037 035032
016010 002370
016012 003434
016014 000400
016016 016022
016020 016026

JSR RO.3#COMPAR
WRFROM
REINTO
256.
7\$
8\$

: COMPARE TWO BLOCKS OF MEMORY
: GOOD DATA STARTS FROM WRFROM
: TEST DATA STARTS FROM REINTO
: 256. WORDS TO BE COMPARED
: RETURN TO 7\$ ON ERROR
: RETURN TO 8\$ ON NO ERROR

016022 104034
016024 000207

7\$:

ERROR 34
RTS PC

: DATA READ FROM A FORCED
: HEADER COMPARE ERROR IS
: INCORRECT
: GOOD DATA GIVES WHAT
: THE READ HEADER AND DATA
: SHOULD HAVE READ
: BAD DATA GIVES WHAT
: WAS IN BUFFER AFTER
: READ COMMAND

016026

8\$:

4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071

```

*****
*TEST 16      HEADER COMPARE ERROR - RHER1 HCE
*****
*          WITH THE HEADS ON CYLINDER 0 A SEARCH COMMAND IS GIVEN
*          FOR CYLINDER 0 TRACK 0 SECTOR 1, ALTHOUGH THE HEADER
*          FOR THIS SECTOR IS CHANGED TO SECTOR 0 HCE-BIT #7
*          IN RHER1 SHOULD NOT SET
*          BECAUSE SEARCH DOES NOT READ HEADER BUT ONLY USES SECTOR COUNTER
*****
†ST16:  SCOPE
        MOV      #STACK.SP      ;RESET STACK
        MOV      #16,‡†STNM     ;SAVE TEST NUMBER
        JSR      PC,‡#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                                   ;R3-RHDS1, R4-RHER1
                                   ;GIVE RH-11 INITIALIZE
                                   ;SETUP UNIT NUMBER
        ;*GET HEADS TO CYLINDER 0
        JSR      PC,‡#CHECKT    ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
                                   ;AND THAT ALL STATUS BITS ARE = 0
        TYPE     .CPHALT        ;CANNOT CONTINUE TESTING IF NOT
        HALT                                     ;STOP TEST
        MOV      ‡#RP4VEC,‡RPVEC ;SET RPO4 VECTOR ADDRESS
                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                   ;'TIME' WILL ONLY SAVE
                                   ;CURRENT CYLINDER ADDRESS
                                   ;AND LOOK AHEAD REGISTERS
        MOV      ‡#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
        BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
                                   ;ENABLE INTERRUPT
        MOV      (SP)+,‡RHCS1    ;GO WITH
                                   ;6 IN RHCS1 FOR RECALIBRATE
                                   ;WITH INTERRUPT ENABLED
        MOV      ‡R1,R0         ;SAVE RHCS1 DURING ABOVE OPERATION
        MOV      ‡R3,R5         ;SAVE RHDS1 DURING ABOVE OPERATION
        WAT                                     ;WAIT FOR DRY BIT TO SET
        RHDS1                                     ;WAIT FOR RHDS1 REGISTER
        DRY                                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
        5600.                                  ;ALLOW 56000 MICRO SECONDS
        5599.                                  ;DRY MUST SET BETWEEN
                                   ;10 AND 111990 MICRO SECONDS
        JSR      PC,‡#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                                   ;R3-RHDS1, R4-RHER1

```

```

016026 000004
016030 012706 001000
016034 012737 000016 004504
016042 004737 033004
016046 004737 033064
016052 104400 056662
016056 000000
016060 013777 004506 134100
016066 013746 002326
016072 052716 000101
016076 012677 164076
016102 011100
016104 011305
016106 104412
016110 002222
016112 000200
016114 012740
016116 012737
016120 004737 033004

```

```

4072                                     ;GIVE RH-11 INITIALIZE
4073                                     ;SETUP UNIT NUMBER
4074
4075                                     ;*FILL REGISTERS FOR SEARCH
4076
4077 016124 004037 032734 JSR RO,@SRCH ;SEARCH FOR
4078 016130 000000 0 ;CYLINDER 0
4079 016132 001 ;SECTOR 1
4080 016133 000 ;TRACK 0
4081
4082
4083                                     ;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH
4084 016134 004037 033152 JSR RO,@SAVER ;SAVE REGISTERS
4085 016140 002172 RHCW ;RHCW IS THE FIRST REGISTER SAVED
4086 016142 004512 SAVERE ;STARTING ADDRESS OF WHERE
4087 ;THE REGISTERS ARE SAVED
4088 016144 000022 18. ;NUMBER OF REGISTERS
4089 ;SAVED = 18.
4090
4091
4092 016146 004737 033064 JSR PC,@CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4093 ;AND THAT ALL STATUS BITS ARE = 0
4094 016152 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
4095 016156 000000 HALT ;STOP TEST
4096
4097 016160 013777 004506 164000 MOV @#RPHVEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
4098 ;TO 'TIME1' IF P-CLOCK IS PRESENT
4099 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
4100 ;'TIME' WILL ONLY SAVE
4101 ;CURRENT CYLINDER ADDRESS
4102 ;AND LOOK AHEAD REGISTERS
4103
4104
4105 016166 013746 002334 MOV @#SERCH,-(SP) ;GET READY TO MOVE COMMAND
4106 016172 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
4107 ;ENABLE INTERRUPT
4108 016176 012677 163776 MOV (SP)+,@RHCS1 ;GO WITH
4109 ;WITH INTERRUPT ENABLED
4110 016202 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4111 016204 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4112
4113
4114 016206 104412 WAT ;WAIT FOR DRY BIT TO SET
4115 016210 002222 RHDS1 ;WAIT FOR RHDS1 REGISTER
4116 016212 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4117 016214 001614 908. ;ALLOW 9080 MICRO SECONDS
4118 016216 001507 839. ;DRY MUST SET BETWEEN
4119 ;690 AND 17470 MICRO SECONDS
4120
4121
4122                                     ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4123                                     ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
4124 016220 013746 002334 MOV @#SERCH,-(SP) ;SAVE COMMAND
4125 016224 052716 004301 BIS #IE!GO!DVA!RDY,(SP) ;INCLUDE IE!GO!DVA!RDY
    
```

K11

HEADER COMPARE ERROR - RHER1 HCE

```

4126 016230 011637 001124      MOV      (SP),2*SGDDAT      ;SAVE FOR PRINTOUT
4127 016234 022600                CMP      (SP)+,R0          ;DURING ABOVE OPERATION ONLY IE!GO!DVA!RDY
4128                                ;AND COMMAND SHOULD BE SET
4129 016236 001405                BEQ      67$              ;BRANCH IF GOOD
4130 016240 010037 001126      MOV      R0,2*SBDDAT      ;BAD DATA
4131 016244 010137 004500      MOV      R1,2*REGADR      ;FAILING REGISTER RHCS1
4132 016250 104021                ERROR    21              ;DURING ABOVE OPERATION ONLY
4133                                ;COMMAND AND IE!GO!DVA!RDY SHOULD BE SET
4134 016252 012746 010500      67$:  MOV      #MOL!DPR!VV -(SP) ;SAVE BITS SET DURING OPERATION IN RHCS1
4135 016256 011637 001124      MOV      (SP),2*SGDDAT    ;SAVE FOR PRINTOUT
4136 016262 022605                CMP      (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
4137                                ;SHOULD BE SET
4138 016264 001405                BEQ      69$              ;BRANCH IF GOOD
4139 016266 010537 001126      MOV      R5,2*SBDDAT      ;BAD DATA
4140 016272 010337 004500      MOV      R3,2*REGADR      ;FAILING REGISTER RHDS1
4141 016276 104063                ERROR    63              ;DURING ABOVE OPERATION ONLY
4142                                ;MOL!DPR!VV SHOULD BE SET
4143 016300                        69$:
4144
4145                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
4146
4147 016306 004037 033674      JSR      R0,2*CHREG        ;CHANGE BITS IN SAVED REGISTER
4148 016304 002200                RHCS1                    ;CHANGE RHCS1 REGISTER
4149
4150 016306 000001                1                        ;1 BIT/BITS TO BE CHANGED
4151 016310 000001                1                        ;NEW VALUE OF SC IS 1
4152 016312 100000                SC                        ;CHANGE SC BIT
4153
4154 016314 004037 033674      JSR      R0,2*CHREG        ;CHANGE BITS IN SAVED REGISTER
4155 016320 002222                RHDS1                    ;CHANGE RHDS1 REGISTER
4156
4157 016322 000001                1                        ;1 BIT/BITS TO BE CHANGED
4158 016324 000001                1                        ;NEW VALUE OF ATA IS 1
4159 016326 100000                ATA                       ;CHANGE ATA BIT
4160
4161 016330 053737 004644 004536  BIS      2*ATTENT,2*SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4162                                ;FOR WORKING DRIVE IN
4163                                ;SAVED RHAS LOCATION
4164
4165                                ;*COMPARE REGISTERS BEFORE SEARCH WITH AFTER SEARCH
4166
4167 016336 004037 034002      JSR      R0,2*COMREG      ;COMPARE SAVED REGISTERS WITH
4168                                ;PRESENT VALUE
4169 016342 004512                SAVERE                   ;GOOD DATA SAVED IN 'SAVERE'
4170 016344 002254                WC                        ;TEST DATA STARTING FROM 'RHWC'
4171 016346 000022                18.                      ;18. REGISTERS TO BE COMPARED
4172 016350 016354                1$                       ;RETURN TO 1$ ON ERROR
4173 016352 016360                2$                       ;RETURN TO 2$ ON NO ERROR
4174
4175
4176 016354 104047                1$:  ERROR    47          ;SEARCH TO A NON-EXISTANT
4177
4178 016356 000207                RTS      PC              ;SECTOR CAUSED IMPROPER
4179                                ;REGISTER CHANGE

```

L11

MNDEC-11-DZRJJ-A, RFO4/5/6 FUNCT. CONT. TST-PT 2
DZRJJA.P11 116 HEADER COMPARE ERROR - RHER1 HCE

MACY11 27(655) 30-MAR-76 18:59 PAGE 92

SEQ 0140

4180
4181
4182
4183
4184
4185
4186

016360

25:

;GOOD DATA GIVES WHAT SHOULD
;BE THERE
;RECEIVED DATA GIVES
;WHAT WAS THERE AFTER
;SEARCH

4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240

::*****
:*TEST 17 RESTORE SECTOR 1 CYLINDER 1 TRACK 1

::* THIS REPLACES REMOVED SECTOR

::* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
:* TRACK 0, SECTOR 1, KEYS=0, NUMBER OF WORDS 256 WORDS
:* OF 0

::* THEN READ HEADER AND DATA FOR ABOVE.
:* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
:* 10000, 1, 0, 0, AND 256 OF 0
:* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
:* THE GO BIT, AND ALL THE REGISTERS ARE SAVED
:* THEN GO IS GIVEN FOR WRITE HEADER AND DATA

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

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

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

::*****

4220	016360	000004		
4221	016362	012706	001000	
4222	016366	012737	000017	004504
4223				
4224	016374	004737	033004	
4225				
4226				
4227				
4228				
4229				
4230				
4231	016400	004037	032630	
4232	016404	002370		
4233	016406	000004		
4234	016410	010000		
4235	016412	000001		
4236	016414	000000		
4237	016416	000000		
4238				
4239				
4240				

```

TST17: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #17,2#TSTNM ;SAVE TEST NUMBER

JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

;*FILL WRITE FROM BUFFER WITH HEADER

JSR RD,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
1 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

;*FILL WRITE FROM BUFFER WITH DATA

```



```

4241 016420 004037 032654 JSR RO,2#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
4242 016424 002400 WRFROM+10 ;STARTING FROM WRFROM+10
4243 015426 000400 256. ;256. WORDS
4244 016430 000000 0 ;FILL WITH 0
4245
4246
4247
4248 ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
4249 ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
4250 ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
4251 ;*CHANGE WRITE FROM BUFFER
4252
4253 016432 004037 032630 JSR RO,2#FLHEAD ;SAVE HEADER DATA IN REINTO
4254 016436 003434 REINTO ;LOCATION WHERE SAVED
4255 016440 000004 4 ;NUMBER OF WORDS SAVED
4256 016442 010000 10000 ;FIRST DATA WORD
4257 016444 000001 1 ;SECOND DATA WORD
4258 016446 000000 0 ;THIRD DATA WORD
4259 016450 000000 0 ;FOURTH DATA WORD
4260
4261 016452 004037 032654 JSR RO,2#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10
4262 016456 003444 REINTO+10 ;STARTING FROM REINTO+10
4263 016460 000400 256. ;256. WORDS
4264 016462 000000 0 ;FILL WITH 0
4265
4266
4267 ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
4268
4269 016464 004037 034766 JSR RO,2#RUN ;SETUP TO RUN FOR DATA COMMAND
4270 016470 000000 0 ;CYLINDER 0
4271 016472 001 .BYTE 1 ;SECTOR 1
4272 016473 000 .BYTE 0 ;TRACK 0
4273 016474 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
4274 ;4 HEADER WORDS
4275 016476 002370 WRFROM ;BUS ADDRESS
4276 ;STARTING ADDRESS OF DATA
4277 ;BUFFER = WRFROM
4278 016500 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4279 016502 010000 FMT22 ;16 BITS PER WORD FORMAT
4280 ;DO NOT INHIBIT ECC CORRECTION
4281 ;DO NOT INHIBIT HEADER COMPARE
4282 016504 002344 WRIFOR ;GET READY TO DO A WRIFOR
4283 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
4284
4285
4286 ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
4287 016506 004037 033152 JSR RO,2#SAVER ;SAVE REGISTERS
4288 016512 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
4289 016514 004512 SAVERE ;STARTING ADDRESS OF WHERE
4290 ;THE REGISTERS ARE SAVED
4291 016516 000021 17. ;NUMBER OF REGISTERS
4292 ;SAVED = 17.
4293
4294 016520 004737 033064 JSR PC,2#CHECKT ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1

```



```

4.2 016524 104400 056662 TYPE ,CPHALT ;AND THAT ALL STATUS BITS ARE = 0
4.3 016530 000000 HALT ;CANNOT CONTINUE TESTING IF NOT
4.4 ;STOP TEST
4.5 016532 013777 004506 163426 MOV 3#RPHVEC,3#RVEC ;SET RPH4 VECTOR ADDRESS
4.6 ;TO 'TIME1' IF P-CLOCK IS PRESENT
4.7 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
4.8 ;'TIME' WILL ONLY SAVE
4.9 ;CURRENT CYLINDER ADDRESS
4.10 ;AND LOOK AHEAD REGISTERS
4.11
4.12 016540 013746 002344 MOV 3#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
4.13 016544 052716 003101 BIS 3#GO!IE,(SP) ;GET READY TO SET GO AND
4.14 ;ENABLE INTERRUPT
4.15 016550 012677 163424 MOV (SP)+,3#RHCS1 ;GO WITH
4.16 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
4.17 ;WITH INTERRUPT ENABLED
4.18 016554 011100 MOV 3#R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
4.19 016556 011305 MOV 3#R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
4.20 ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
4.21
4.22 016560 104412 WAIT ;WAIT FOR RDY BIT TO SET
4.23 016562 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
4.24 016564 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4.25 016566 001614 909. ;ALLOW 9080 MICRO SECONDS
4.26 016570 001507 839. ;RDY MUST SET BETWEEN
4.27 ;690 AND 17470 MICRO SECONDS
4.28
4.29 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4.30 ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
4.31
4.32 016572 013746 002344 MOV 3#WRIFOR,-(SP) ;SAVE COMMAND
4.33 016576 052716 004101 BIS 3#IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
4.34 016602 011637 001124 MOV (SP),3#SGDDAT ;SAVE FOR PRINTOUT
4.35 016606 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
4.36 ;AND COMMAND SHOULD BE SET
4.37 016610 001405 BEQ 64$ ;BRANCH IF GOOD
4.38 016612 010037 001126 MOV R0,3#SBDDAT ;BAD DATA
4.39 016616 010137 004500 MOV R1,3#REGADR ;FAILING REGISTER RHCS1
4.40 016622 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
4.41 ;COMMAND AND IE!GO!DVA SHOULD BE SET
4.42 64$: 016624 012746 010500 MOV 3#MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHCS1
4.43 016630 011637 001124 MOV (SP),3#SGDDAT ;SAVE FOR PRINTOUT
4.44 016634 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
4.45 ;SHOULD BE SET
4.46 016636 001405 BEQ 66$ ;BRANCH IF GOOD
4.47 016640 010537 001126 MOV R5,3#SBDDAT ;BAD DATA
4.48 016644 010337 004500 MOV R3,3#REGADR ;FAILING REGISTER RHDS1
4.49 016650 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4.50 ;MOL!DPR!VV SHOULD BE SET
4.51 66$:
4.52 016652
4.53 ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
    
```

016652
016656
016660
016664
016668
016672
016676
016680
016684
016688
016692
016696
016700
016704
016708
016712
016716
016720
016724
016728
016732
016736
016740
016744

```

016652 004037 032706      JSR    RD,0#FILLRE      ;MOV 0 INTO SAVED RHWC
016656 002172                          RHWC                       ;SAVED REGISTER TO CHANGE
016660 000000                          0                           ;DATA

016664 004037 032706      JSR    RD,0#FILLRE      ;MOV WRFROM+(260.*2) INTO SAVED RHBA
016668 002174                          RHBA                       ;SAVED REGISTER TO CHANGE
016672 000400                          WRFROM+(260.*2)           ;DATA

016676 004037 032706      JSR    RD,0#FILLRE      ;MOV 2 INTO SAVED RHDST
016680 002204                          RHDST                      ;SAVED REGISTER TO CHANGE
016684 000002                          2                           ;DATA

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

016702 004037 034002      JSR    RD,0#COMREG      ;COMPARE SAVED REGISTERS WITH
016706 004512                          SAVERE                     ;PRESENT VALUE
016710 002354                          WC                          ;GOOD DATA SAVED IN 'SAVERE'
016712 000001                          17                          ;TEST DATA STARTING FROM 'RHWC'
016714 016720                          17                          ;17. REGISTERS TO BE COMPARED
016716 016724                          15                          ;RETURN TO 15 ON ERROR
                                  25                          ;RETURN TO 25 ON NO ERROR

016720 104027 35:        ERROR   27                ;WRITE HEADER AND DATA
016724 000207                          RTS    PC                   ;CAUSED IMPROPER REGISTER
                                  ;CHANGE
                                  ;GOOD DATA GIVES WHAT SHOULD
                                  ;BE THERE
                                  ;RECEIVED DATA GIVES WHAT
                                  ;WAS THERE AFTER COMMAND

                                  ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
                                  ;*NOTHER GOT CHANGED

016724 35:
016724 004037 035032      JSR    RD,0#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
016730 003434                          REINTO                     ;GOOD DATA STARTS FROM REINTO
016732 002370                          WRFROM                     ;TEST DATA STARTS FROM WRFROM
016734 000404                          260.                       ;260. WORDS TO BE COMPARED
016736 016742                          35                          ;RETURN TO 35 ON ERROR
016740 016746                          45                          ;RETURN TO 45 ON NO ERROR

016742 104030 35:        ERROR   30                ;WRITE HEADER AND DATA
016744 000207                          RTS    PC                   ;CHANGED WRITE FROM BUFFER

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

```

016746
016746
016752
016756
016760
016762
016764
016770
016772
016773
016774
016776
017000
017002
017004
017006
017012
017014
017016
017020
017024
017030
017032
017040

016746
016746 004737 033004
016752 004037 032654
016756 003434
016760 003434
016762 177777

016764 004037 034766
016770 000000
016772 001
016773 000
016774 177374
016776 003434

017000 000000
017002 014000

017004 002350

017006 004037 033152
017012 002172
017014 004512
017016 000022

017020 004737 033064
017024 104400 056662
017030 000000

017032 013777 004506 163126

017040 013746 002350

4S:
JSR PC,2#CLDISK
JSR RD,2#CLAREA
REINTO 260.
-1

;*NOW FILL COMMAND
JSR RD,2#RUN
0
.BYTE 1
.BYTE 0
-256.-4
REINTO
0
ECI!FMT22
REFOR

;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
JSR RD,2#SAVER
RHWC
SAVERE
18.
JSR PC,2#CHECKT
TYPE ,CPHALT
HALT
MOV 2#RPHVEC,2#RPVEC
MOV 2#REFOR,-(SP)

: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

:SETUP TO RUN FOR DATA COMMAND
:CYLINDER 0
:SECTOR 1
:TRACK 0
:WORD COUNT (DATA) = 256. +
:4 HEADER WORDS
:BUS ADDRESS
:STARTING ADDRESS OF DATA
:BUFFER = REINTO
:DO NOT INHIBIT BUS ADDRESS INCREMENT
:16 BITS PER WORD FORMAT
:INHIBIT ECC CORRECTION
:DO NOT INHIBIT HEADER COMPARE
:GET READY TO DO A REFOR
:READ HEADER AND DATA WITH 72 IN RHCS1

:SAVE REGISTERS
:RHWC IS THE FIRST REGISTER SAVED
:STARTING ADDRES OF WHERE
:THE REGISTERS ARE SAVED
:NUMBER OF REGISTERS
:SAVED = 18.

:CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
:AND THAT ALL STATUS BITS ARE = 0
:CANNOT CONTINUE TESTING IF NOT
:STOP TEST

:SET RPO4 VECTOR ADDRESS
:TO 'TIME1' IF P-CLOCK IS PRESENT
:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
:'TIME' WILL ONLY SAVE
:CURRENT CYLINDER ADDRESS
:AND LOOK AHEAD REGISTERS

:GET READY TO MOVE COMMAND

```

4457 017044 052716 000101      BIS      #GO!IE,(SP)      ;GET READY TO SET GO AND
4458 017050 012677 163124      MOV      (SP)+,RHC1    ;ENABLE INTERRUPT
4459 017054 011100      MOV      R1,R0        ;GO WITH
4460 017056 011305      MOV      R3,R5        ;:72 IN RHC1 FOR READ DATA
4461 017060 104412      WAT                      ;WITH INTERRUPT ENABLED
4462 017062 002200      RHC1                    ;SAVE RHC1 DURING ABOVE OPERATION
4463 017064 000200      RDY                      ;SAVE RHDS1 DURING ABOVE OPERATION
4464 017066 001614      908.                    ;WAIT FOR RDY BIT TO SET
4465 017070 001507      839.                    ;WAIT FOR RHC1 REGISTER
4466 017072 013746 002350      ;*COMPARE CONTENTS OF RHC1 AND RHDS1 ALREADY SAVED IN
4467 017076 052716 004101      ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
4468 017102 011637 001124      MOV      R0,R0        ;SAVE COMMAND
4469 017106 022600      CMP      (SP)+,R0     ;INCLUDE IE!GO!DVA (SP)
4470 017110 001405      BEQ      67$          ;SAVE FOR PRINTOUT
4471 017112 010037 001126      MOV      R0,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
4472 017116 010137 004500      MOV      R1,R0        ;AND COMMAND SHOULD BE SET
4473 017122 104021      ERROR   21           ;BRANCH IF GOOD
4474 017124 012746 010500      67$: MOV      #MOL!DPR!VV,(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4475 017130 011637 001124      MOV      (SP),R0     ;SAVE FOR PRINTOUT
4476 017134 022605      CMP      (SP)+,R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
4477 017136 001405      BEQ      69$          ;SHOULD BE SET
4478 017140 010537 001126      MOV      R5,R0        ;BRANCH IF GOOD
4479 017144 010337 004500      MOV      R3,R0        ;BAD DATA
4480 017150 104063      ERROR   63           ;FAILING REGISTER RHDS1
4481 017152 017152      69$:                    ;DURING ABOVE OPERATION ONLY
4482 017156 004037 032706      JSR      R0,R0        ;MOL!DPR!VV SHOULD BE SET
4483 017160 002172 000000      RHW0                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4484 017162 004037 032706      JSR      R0,R0        ;MOV 0 INTO SAVED RHW0
4485 017166 002174 004444      RHW0                    ;SAVED REGISTER TO CHANGE
4486 017170 004444 000000      REINTO+(260.*2)      ;DATA
4487 017172 004037 032706      JSR      R0,R0        ;MOV REINTO+(260.*2) INTO SAVED RHBA
4488 017176 002204 000000      RHW0                    ;SAVED REGISTER TO CHANGE
4489 017176 002204 000000      RHW0                    ;DATA

```

```

017200 000002 2 ;DATA
;
;*COMPARE REGISTER BEFORE READ HEADER AND DATA
;*WITH REGISTERS AFTER COMMAND

017202 004037 034002 JSR RC,0#COMREG ;COMPARE SAVED REGISTERS WITH
;PRESENT VALUE
017206 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
017210 002354 WC ;TEST DATA STARTING FROM 'RHW'
017212 000022 18. ;18. REGISTERS TO BE COMPARED
017214 017220 5$ ;RETURN TO 5$ ON ERROR
017216 017224 6$ ;RETURN TO 6$ ON NO ERROR

017220 104031 5$: ERROR 31 ;READ HEADER AND DATA CAUSED
017222 000237 RTS PC ;IMPROPER REGISTER CHANGE
;GOOD DATA GIVES WHAT SHOULD
;BE THERE
;RECEIVED DATA GIVES WHAT WAS
;THERE AFTER COMMAND

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

017224 6$:

017224 004037 035032 JSR RC,0#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
017230 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
017232 003434 REINTO ;TEST DATA STARTS FROM REINTO
017234 000404 260. ;260. WORDS TO BE COMPARED
017236 017242 7$ ;RETURN TO 7$ ON ERROR
017240 017246 10$ ;RETURN TO 10$ ON NO ERROR

017242 104032 7$: ERROR 32 ;WRITE HEADER AND DATA
017244 000207 RTS PC ;FOLLOWED BY A READ HEADER
;AND DATA GAVE A READ ERROR
;ERROR MAY BE IN READ OR WRITE

017246 10$:

```

45755
45756
45757
45758
45759
45760
45761
45762
45763
45764
45765
45766
45767
45768
45769
45770
45771
45772
45773
45774
45775
45776
45777
45778
45779
45780
45781
45782
45783
45784
45785
45786
45787
45788
45789
45790
45791
45792
45793
45794
45795
45796
45797
45798
45799
45800
45801
45802
45803
45804
45805
45806
45807
45808
45809
45810
45811
45812
45813
45814
45815
45816
45817
45818
45819
45820

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

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

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

TST20: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #20, @TSTNM ;SAVE TEST NUMBER
JSR PC, @CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

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

TST @RPO6 ;TEST FOR RPO6 DRIVE
BEQ SS ;TREAT UNIT AS AN RPO4
;TREAT AS AN RPO6

017274 004037 032630 JSR R0, @FLHEAD ;SAVE HEADER DATA IN WRFROM
017300 002370 WRFROM ;LOCATION WHERE SAVED
017302 000004 4 ;NUMBER OF WORDS SAVED
017304 011457 11457 ;FIRST DATA WORD
017306 000000 0 ;SECOND DATA WORD
017310 000000 0 ;THIRD DATA WORD
017312 000000 0 ;FOURTH DATA WORD
017314 000410 BR 68 ;CONTINUE WITH SET UP

SS:

017316 004037 032630 JSR R0, @FLHEAD ;SAVE HEADER DATA IN WRFROM
017322 002370 WRFROM ;LOCATION WHERE SAVED
017324 000004 4 ;NUMBER OF WORDS SAVED
017326 010633 10633 ;FIRST DATA WORD
017330 000000 0 ;SECOND DATA WORD
017332 000000 0 ;THIRD DATA WORD
017334 000000 0 ;FOURTH DATA WORD
017336 BR 68 ;CONTINUE

68:

*FILL WRITE FROM BUFFER WITH DATA

017336 004037 032654 JSR R0, @CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
017342 002400 WRFROM+10 ;STARTING FROM WRFROM+10
017344 000400 256. ;256. WORDS

4609 017346 177777

0-1

;FILL WITH 0-1

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

4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662

017350 005737 004636
017354 001412

::*****
TST 3#RPO6 ;TEST FOR RPO6 DRIVE
BEQ 35 ;TREAT UNIT AS RPO4
;TREAT UNIT AS RPO6

017356 004037 034766
017362 001457
017364 000
017365 000
017366 177374

JSR RO,3#RUN
815.
.BYTE 0
.BYTE 0
-256.-4

;SETUP TO RUN FOR DATA COMMAND
;CYLINDER 815.
;SECTOR 0
;TRACK 0
;WORD COUNT (DATA) = 256. +
;4 HEADER WORDS
;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM
;DO NOT INHIBIT BUS ADDRESS INCREMENT
;16 BITS PER WORD FORMAT
;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A WRIFOR
;WRITE HEADER AND DATA WITH 62 IN RHCS1

017370 002370

WRFROM

017372 000000
017374 010000

0
FMT22

017376 002344

WRIFOR

;WRITE HEADER AND DATA WITH 62 IN RHCS1

017400 000411
017402

35:

BR 45

;CONTINUE WITH TESTING

017402 004037 034766
017406 000633
017410 000
017411 000
017412 177374

JSR RO,3#RUN
411.
.BYTE 0
.BYTE 0
-256.-4

;SETUP TO RUN FOR DATA COMMAND
;CYLINDER 411.
;SECTOR 0
;TRACK 0
;WORD COUNT (DATA) = 256. +
;4 HEADER WORDS
;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM
;DO NOT INHIBIT BUS ADDRESS INCREMENT
;16 BITS PER WORD FORMAT
;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A WRIFOR
;WRITE HEADER AND DATA WITH 62 IN RHCS1

017414 002370

WRFROM

017416 000000
017420 010000

0
FMT22

017422 002344

WRIFOR

;WRITE HEADER AND DATA WITH 62 IN RHCS1

017424

45:

;CONTINUE


```

4717
4718 017534 004037 033674 JSR RO, @#CHREG ;CHANGE BITS IN SAVED REGISTER
4719 017540 002222 RHDS1 ;CHANGE RHDS1 REGISTER
4720
4721 017542 000002 2 ;2 BIT/BITS TO BE CHANGED
4722 017544 000001 1 ;NEW VALUE OF ATA IS 1
4723 017546 100000 ATA ;CHANGE ATA BIT
4724 017550 000001 1 ;NEW VALUE OF ERR IS 1
4725 017552 040000 ERR ;CHANGE ERR BIT
4726
4727 017554 004037 033674 JSR RO, @#CHREG ;CHANGE BITS IN SAVED REGISTER
4728 017550 002202 RHER1 ;CHANGE RHER1 REGISTER
4729
4730 017562 000001 1 ;1 BIT/BITS TO BE CHANGED
4731 017564 000001 1 ;NEW VALUE OF IAE IS 1
4732 017566 002000 IAE ;CHANGE IAE BIT
4733
4734 017570 053737 004644 004536 BIS @#ATTENT, @#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4735 ;FOR WORKING DRIVE IN
4736 ;SAVED RHAS LOCATION
4737 017576 017737 162402 004524 MOV @RHDS1, @#SAVERE+12 ;RHDS1 IS INDETERMINATE
4738 ;SO IT IS NOT CHECKED
4739
4740
4741 ;*COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH
4742 ;*CONTENTS AFTER ATTEMPTED WRITE WITH AN 'IAE' ERROR
4743 ;:*****
4744
4745
4746 017604 004037 034002 JSR RO, @#COMREG ;COMPARE SAVED REGISTERS WITH
4747 ;PRESENT VALUE
4748 017610 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4749 017612 002254 WC ;TEST DATA STARTING FROM 'RHWC'
4750 017614 000022 18. ;18. REGISTERS TO BE COMPARED
4751 017616 017622 1$ ;RETURN TO 1$ ON ERROR
4752 017620 017626 2$ ;RETURN TO 2$ ON NO ERROR
4753
4754
4755 017622 104054 1$: ERROR 54 ;ATTEMPTED OPERATION WITH
4756 017624 000207 RTS PC ;INVALID ADDRESS CAUSED
4757 ;IMPROPER REGISTER CHANGE
4758 ;GOOD DATA GIVES WHAT SHOULD
4759 ;BE THERE
4760 017626 2$: ;RECEIVED DATA GIVES REGISTER
4761 ;CONTENTS AFTER ATTEMPTED
4762 ;WRITE HEADER AND DATA
4763

```



```

4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774 017626 000004
4775 017630 012706 001000
4776 017634 012737 000021 004504
4777
4778 017642 004737 033004
4779
4780
4781
4782
4783
4784
4785 017646 004037 032654
4786 017652 002370
4787 017654 000400
4788 017656 000377
4789
4790
4791
4792
4793 017660 004037 034766
4794 017664 000000
4795 017666 000
4796 017667 023
4797 017670 177400
4798 017672 002370
4799
4800
4801 017674 000000
4802 017676 010000
4803
4804
4805 017700 002342
4806
4807
4808
4809
4810 017702 004037 033152
4811 017706 002172
4812 017710 004512
4813
4814 017712 000022
4815
4816
4817 017714 004737 033064

```

```

:*****
:*TEST 21      INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE
:*
:*      A WRITE DATA IS ATTEMPTED TO CYLINDER 0, TRACK 19,
:*      SECTOR 0
:*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
:*      SHOULD SET
:*****
†ST21: SCOPE
MOV      #STACK,SP      ;RESET STACK
MOV      #21,‡#TSTNM    ;SAVE TEST NUMBER
JSR      PC,‡#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                        ;R3-RHDS1, R4-RHER1
                        ;GIVE RH-11 INITIALIZE
                        ;SETUP UNIT NUMBER
; *FILL WRITE FROM BLFFER WITH DATA
JSR      RO,‡#CLAREA    ;CLEAR 256. WORDS. FROM WRFROM
WRFROM   ;STARTING FROM WRFROM
256.     ;256. WORDS
377      ;FILL WITH 377
; *WRITE DATA COMMAND WILL BE FILLED
JSR      RO,‡#RUN       ;SETUP TO RUN FOR DATA COMMAND
0        ;CYLINDER 0
.BYTE 0   ;SECTOR 0
.BYTE 19. ;TRACK 19.
-256.    ;WORD COUNT = 256.
WRFROM   ;BUS ADDRESS
         ;STARTING ADDRESS OF DATA
         ;BUFFER = WRFROM
0        ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22    ;16 BITS PER WORD FORMAT
         ;DO NOT INHIBIT ECC CORRECTION
         ;DO NOT INHIBIT HEADER COMPARE
WRIDAT   ;GET READY TO DO A WRIDAT
         ;WRITE DATA WITH 60 IN RHCS1
; *SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
JSR      RO,‡#SAVER     ;SAVE REGISTERS
RHWC     ;RHWC IS THE FIRST REGISTER SAVED
SAVERE   ;STARTING ADDRES OF WHERE
         ;THE REGISTERS ARE SAVED
18.     ;NUMBER OF REGISTERS
         ;SAVED = 18.
JSR      PC,‡#CHECKT    ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1

```

```

4818                                     ;AND THAT ALL STATUS BITS ARE = C
4819 017720 104400 056662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
4820 017724 000000 HALT ;STOP TEST
4821
4822 017726 013777 004506 162232 MOV @RPO4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
4823                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
4824                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
4825                                     ;'TIME' WILL ONLY SAVE
4826                                     ;CURRENT CYLINDER ADDRESS
4827                                     ;AND LOOK AHEAD REGISTERS
4828
4829
4830 017734 013746 002342 MOV @WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
4831 017740 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
4832                                     ;ENABLE INTERRUPT
4833 017744 012677 162230 MOV (SP)+,@RHCS1 ;GO WITH
4834                                     ;60 IN RHCS1 FOR WRITE DATA
4835                                     ;WITH INTERRUPT ENABLED
4836
4837
4838 017750 104412 WAT ;WAIT FOR IAE BIT TO SET
4839 017752 002202 RHER1 ;WAIT FOR RHER1 REGISTER
4840 017754 002000 IAE ;WAIT FOR IAE BIT IN RHER1 REGISTER
4841 017756 000011 9. ;ALLOW 90 MICRO SECONDS
4842 017760 000011 9. ;IAE MUST SET BETWEEN
4843                                     ;00 AND 180 MICRO SECONDS
4844
4845 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4846
4847
4848
4849
4850 ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
4851 ;*RHWC,RHBA,RHCS1,RHCS2, CANNOT BE PEREDETERMINED
4852 ;*THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
4853 017762 017737 162204 004512 MOV @RHWC,@SAVERE ;RHWC IS UNPREDICTABLE
4854                                     ;AS EXPLAINED ABOVE
4855 017770 017737 162200 004514 MOV @RHBA,@SAVERE+2 ;RHBA IS UNPREDICTABLE
4856                                     ;AS EXPLAINED ABOVE
4857 017776 017737 162174 004516 MOV @RHCS2,@SAVERE+4 ;RHCS2 IS UNPREDICTABLE
4858                                     ;AS EXPLAINED ABOVE
4859 020004 017737 162170 004520 MOV @RHCS1,@SAVERE+6 ;RHCS1 IS UNPREDICTABLE
4860                                     ;AS EXPLAINED ABOVE
4861
4862 020012 004037 033674 JSR RC,@CHREG ;CHANGE BITS IN SAVED REGISTER
4863 020016 002222 RHDS1 ;CHANGE RHDS1 REGISTER
4864
4865 020020 000002 2 ;2 BIT/BITS TO BE CHANGED
4866 020022 000001 1 ;NEW VALUE OF ATA IS 1
4867 020024 100000 ATA ;CHANGE ATA BIT
4868 020026 000001 1 ;NEW VALUE OF ERR IS 1
4869 020030 040000 ERR ;CHANGE ERR BIT
4870 020032 017737 162146 004524 MOV @RHDST,@SAVERE+12 ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
4871

```

```

4872
4873 020040 004037 033674 JSR RD,2#CHREG ;CHANGE BITS IN SAVED REGISTER
4874 020044 002202 RHER1 ;CHANGE RHER1 REGISTER
4875
4876 020046 000001 1 ;1 BIT/BITS TO BE CHANGED
4877 020050 000001 1 ;NEW VALUE OF IAE IS 1
4878 020052 002000 IAE ;CHANGE IAE BIT
4879
4880 020054 053737 004644 004536 BIS 2#ATTENT,2#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
4881 ;FOR WORKING DRIVE IN
4882 ;SAVED RHAS LOCATION
4883
4884 ;*COMPARE REGISTERS BEFORE ATTEMPTED WRITE DATA
4885 ;*WITH AFTER ATTEMPT, IAE SHOULD BE SET
4886
4887 020062 004037 034002 JSR RD,2#COMREG ;COMPARE SAVED REGISTERS WITH
4888 ;PRESENT VALUE
4889 020066 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4890 020070 002254 WC ;TEST DATA STARTING FROM 'R.HWC'
4891 020072 000022 18. ;18. REGISTERS TO BE COMPARED
4892 020074 020100 1$ ;RETURN TO 1$ ON ERROR
4893 020076 020104 2$ ;RETURN TO 2$ ON NO ERROR
4894
4895
4896 020100 104054 1$: ERROR 54 ;ATTEMPTED WRITE DATA
4897 020102 000207 RTS PC ;WITH INVALID ADDRESS
4898 ;CAUSED IMPROPER REGISTER
4899 ;CHANGE
4900 020104 2$: ;GOOD DATA GIVES WHAT
4901 ;SHOULD BE THERE
4902 ;RECEIVED DATA GIVES WHAT
4903 ;WAS THERE AFTER AFTER ATTEMPT
4904

```

INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE

4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958

020104 000004
020105 012706 001000
020112 012737 000022 004504
020120 004737 033004
020124 004737 033064
020130 104400 056662
020134 000000
020136 013777 004506 162022
020144 004037 032754
020150 000010
020152 013746 002352
020156 052716 000101
020162 012677 162012
020166 104412
020170 002222
020172 000200
020174 015530

*TEST 22 INVALID ADDRESS ERROR RHER1 -BIT #10

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

TST2: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #22, @#TSTNM ;SAVE TEST NUMBER
JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
;*GET THE HEADS TO CYLINDER 10
JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
HALT ;STOP TEST
MOV @#RPHVEC, @#RPVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS
JSR RO, @#SEEKCY ;SEEK FOR
10 ;CYLINDER 10
MOV @#SEECOM, -(SP) ;GET READY TO MOVE COMMAND
BIS #GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT
MOV (SP)+, @#RHCS1 ;GO WITH
;4 IN RHCS1 FOR SEEK
;WITH INTERRUPT ENABLED
WAT ;WAIT FOR DRY BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7000. ;ALLOW 70000 MICRO SECONDS

DATA.P11

↑22 INVALID ADDRESS ERROR RHER1 -BIT #10

```

4959 020176 015530          7000.          :DRY MUST SET BETWEEN
4960          :00 AND 140000 MICRO SECONDS
4961
4962
4963 020220 004737 033004  JSR      PC,0#CLDISK  :SET R1-RHCS1, R2-RHCS2
4964          :R3-RHDS1, R4-RHER1
4965          :GIVE RH-11 INITIALIZE
4966          :SETUP UNIT NUMBER
4967
4968          :*FILL READ INTO BUFFER WITH 125252
4969
4970 020220 004037 032654  JSR      RO,0#CLAREA  :CLEAR 260 WORDS FROM REINTO
4971          :STARTING FROM REINTO
4972          :260 WORDS
4973          :FILL WITH 125252
4974
4975
4976          :*THE READ HEADER AND DATA COMMAND IS FILLED
4977
4978 020216 004037 034766  JSR      RO,0#RUN     :SETUP TO RUN FOR DATA COMMAND
4979          :CYLINDER 0
4980          :SECTOR 22.
4981          :TRACK 0
4982          :WORD COUNT (DATA) = 256. +
4983          :+ 4 HEADER WORDS
4984          :BUS ADDRESS
4985          :STARTING ADDRESS OF DATA
4986          :BUFFER = REINTO
4987          :DO NOT INHIBIT BUS ADDRESS INCREMENT
4988          :16 BITS PER WORD FORMAT
4989          :INHIBIT ECC CORRECTION
4990          :DO NOT INHIBIT HEADER COMPARE
4991          :GET READY TO DO A REFOR
4992          :READ HEADER AND DATA WITH 72 IN RHCS1
4993
4994          :*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
4995          JSR      RO,0#SA EP  :SAVE REGISTERS
4996          RHWC          :RHWC IS THE FIRST REGISTER SAVED
4997          SAVERE        :STARTING ADDRESS OF WHERE
4998          :THE REGISTERS ARE SAVED
4999          :NUMBER OF REGISTERS
5000          :SAVED = 18.
5001
5002 020240 004037 033152  JSR      RO,0#SA EP
5003 020244 002172
5004 020246 004512
5005
5006 020250 000022          18.
5007
5008 020252 004737 033064  JSR      PC,0#CHECKT  :CHECKS DVA, RDY, MCL, DPR, DRY AND VV = 1
5009          :AND THAT ALL STATUS BITS ARE = 0
5010 020256 104400 056662  TYPE     ,CPHALT     :CANNOT CONTINUE TESTING IF NOT
5011 020262 000000          HALT                :STOP TEST
5012
5013 020264 013777 004506 161674  MOV      0#RP4VEC,0#RPVEC :SET RFD4 VECTOR ADDRESS
5014          :TO 'TIME1' IF P-CLOCK IS PRESENT
5015          :OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5016          :'TIME' WILL ONLY SAVE
5017          :CURRENT CYLINDER ADDRESS

```

D2RJJ.A.P11 t22 INVALID ADDRESS ERROR RHER1 -BIT #10

5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066

```

020372 013746 002350      MOV      @REFOR,-(SP)      ;AND LOOK AHEAD REGISTERS
020376 052716 000101      BIS      @GO!IE,(SP)      ;GET READY TO MOVE COMMAND
                                ;GET READY TO SET GO AND
                                ;ENABLE INTERRUPT
020302 012677 161672      MOV      (SP)+,@RHCS1     ;GO WITH
                                ;72 IN RHCS1 FOR READ DATA
                                ;WITH INTERRUPT ENABLED

020306 134412      WAT                      ;WAIT FOR IAE BIT TO SET
020310 002202      RHER1                    ;WAIT FOR RHER1 REGISTER
020312 002200      IAE                      ;WAIT FOR IAE BIT IN RHER1 REGISTER
020314 000002      2.                       ;ALLOW 20 MICRO SECONDS
020316 000002      2.                       ;IAE MUST SET BETWEEN
                                ;00 AND 40 MICRO SECONDS

                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

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

020320 017737 161646 004512      MOV      @RHWC,@SAVERE    ;RHWC IS UNPREDICTABLE
                                ;AS EXPLAINED ABOVE
020326 017737 161642 004514      MOV      @RHBA,@SAVERE+2  ;RHBA IS UNPREDICTABLE
                                ;AS EXPLAINED ABOVE
020334 017737 161636 004516      MOV      @RHCS2,@SAVERE+4 ;RHCS2 IS UNPREDICTABLE
                                ;AS EXPLAINED ABOVE
020342 017737 161632 004520      MOV      @RHCS1,@SAVERE+6 ;RHCS1 IS UNPREDICTABLE
                                ;AS EXPLAINED ABOVE

020350 004037 033674      JSR      RC,@CHREG        ;CHANGE BITS IN SAVED REGISTER
020354 002222      RHDS1                    ;CHANGE RHDS1 REGISTER

020356 000002      2                        ;2 BIT/BITS TO BE CHANGED
020360 000001      1                        ;NEW VALUE OF ATA IS 1
020362 100000      ATA                      ;CHANGE ATA BIT
020364 000001      1                        ;NEW VALUE OF ERR IS 1
020366 040000      ERR                      ;CHANGE ERR BIT
020370 017737 161610 004524      MOV      @RHDS1,@SAVERE+12 ;RHDS1 IS INDETERMINATE SO IT IS NOT CHECKED

020376 004037 033674      JSR      RC,@CHREG        ;CHANGE BITS IN SAVED REGISTER
020402 002202      RHER1                    ;CHANGE RHER1 REGISTER

020404 000001      1                        ;1 BIT/BITS TO BE CHANGED
020406 000001      1                        ;NEW VALUE OF IAE IS 1
020410 002000      IAE                      ;CHANGE IAE BIT

020412 053737 004644 004536      BIS      @ATTENT,@SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
                                ;FOR WORKING DRIVE IN

```


E13

5093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145

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

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

::*****

020442 000004
020444 012736 001000
020450 012737 000023 004534

020456 004737 033004

020462 004037 032654
020466 002434
020470 000260
020472 125252

020474 004037 034766
020500 000000
020502 024
020503 000
020504 177400
020506 003434

020510 000000
020512 004000

020514 002346

020516 004037 033152
020522 002172
020524 004512

020526 000022

020530 004737 033064

↑ST23: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #23, #STNM ;SAVE TEST NUMBER

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

;*FILL READ INTO BUFFER WITH 125252

JSR RO, #CLAREA ;CLEAR 260 WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
260 ;260 WORDS
125252 ;FILL WITH 125252

;*THE READ HEADER AND DATA COMMAND IS FILLED

JSR RO, #RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 20. ;SECTOR 20.
.BYTE 0 ;TRACK 0
-256. ;WORD COUNT = 256.
REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI ;18 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

;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
↑JSR RO, #SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
18. ;NUMBER OF REGISTERS
;SAVED = 18.

JSR PC, #CHECKY ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1

F13

```

147 020534 104400 056662 TYPE ,CPHALT ;AND THAT ALL STATUS BITS ARE = 0
148 020540 000000 HALT ;CANNOT CONTINUE TESTING IF NOT
149 ;STOP TEST
150
151 020542 013777 004506 161416 MOV @#RPHVEC,@RPHVEC ;SET RPH4 VECTOR ADDRESS
152 ;TO 'TIME1' IF P-CLOCK IS PRESENT
153 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
154 ;'TIME' WILL ONLY SAVE
155 ;CURRENT CYLINDER ADDRESS
156 ;AND LOOK AHEAD REGISTERS
157
158
159 020550 013746 002346 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
160 020554 052716 000101 BIS @GO!IE,(SP) ;GET READY TO SET GO AND
161 ;ENABLE INTERRUPT
162 020560 012677 161414 MOV (SP)+,@RHCS1 ;GO WITH
163 ;70 IN RHCS1 FOR READ DATA
164 ;WITH INTERRUPT ENABLED
165
166
167 020564 104412 ;WAIT FOR IAE BIT TO SET
168 020566 002202 RHER1 ;WAIT FOR RHER1 REGISTER
169 020570 002000 IAE ;WAIT FOR IAE BIT IN RHER1 REGISTER
170 020572 000002 ;ALLOW 20 MICRO SECONDS
171 020574 000002 ;IAE MUST SET BETWEEN
172 ;00 AND 40 MICRO SECONDS
173
174 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
175
176 020576 004037 033674 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
177 020602 002200 RHCS1 ;CHANGE RHCS1 REGISTER
178
179 2 ;2 BIT/BITS TO BE CHANGED
180 1 ;NEW VALUE OF SC IS 1
181 020604 000002 SC ;CHANGE SC BIT
182 1 ;NEW VALUE OF TRE IS 1
183 020606 000001 TRE ;CHANGE TRE BIT
184 020610 100000 ;TRE IS UNPREDICTABLE
185 020612 000001 MOV @RHDST,@#SAVERE+12;RHDST IS UNPREDICTABLE
186 020614 040000
187 020616 017737 161362 004524
188
189 020624 004037 033674 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
190 020630 002222 RHDS1 ;CHANGE RHDS1 REGISTER
191
192 2 ;2 BIT/BITS TO BE CHANGED
193 1 ;NEW VALUE OF ATA IS 1
194 020632 000002 ATA ;CHANGE ATA BIT
195 1 ;NEW VALUE OF ERR IS 1
196 020634 000001 ERR ;CHANGE ERR BIT
197 020636 100000
198 020640 000001
199 020642 040000
200 020644 004037 033674 JSR RO,@#CHREG ;CHANGE BITS IN SAVED REGISTER
201 020650 002202 RHER1 ;CHANGE RHER1 REGISTER
202
203 1 ;1 BIT/BITS TO BE CHANGED
204 1 ;NEW VALUE OF IAE IS 1
205 020652 000001 IAE ;CHANGE IAE BIT
206 020654 000001
207 020656 002000
    
```

000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030

```

020660 053737 004644 004536      BIS      3*ATTENT,2*SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
                                           ;FOR WORKING DRIVE IN
                                           ;SAVED RHAS LOCATION

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

020666 004037 034002      JSR      RD,2*COMREG      ;COMPARE SAVED REGISTERS WITH
                                           ;PRESENT VALUE
020672 004512      SAVERE   ;GOOD DATA SAVED IN 'SAVERE'
020674 002254      WC      ;TEST DATA STARTING FROM 'RHWC'
020676 000022      18.    ;18. REGISTERS TO BE COMPARED
020700 020704      1$    ;RETURN TO 1$ ON ERROR
020702 020710      2$    ;RETURN TO 2$ ON NO ERROR

020704 104051      1$:    ERROR  54 ;ATTEMPTED READ
020706 000207      RTS     PC   ;DATA WITH INVALID
                                           ;ADDRESS CAUSED IMPROPER
                                           ;REGISTER CHANGE
                                           ;GOOD DATA GIVES WHAT
                                           ;SHOULD BE THERE
                                           ;RECEIVED DATA GIVES
                                           ;REGISTERS CONTENTS
                                           ;AFTER ATTEMPTED
                                           ;READ

020710      2$:

```

Handwritten mark resembling a stylized 'J' or 'L'.

3.

8

Handwritten signature or initials.

H13

5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283

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

* A WRITE HEADER AND DATA COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0
 * SECTOR 0, 256 WORDS OF 0
 * NO CHECK IS DONE AFTER THIS WRITE

* A WRITE HEADER AND DATA COMMAND IS GIVEN FOR
 * CYLINDER 410./814., TRACK 18, SECTOR 21, 261 WORDS

* ADDRESS OVERFLOW ERROR - RHER1 BIT#9 (AOE) SHOULD SET
 * AFTER SECTOR 21 IS WRITTEN
 * ALL REGISTERS ARE CHECKED

* A READ HEADER AND DATA CYLINDER 410./814., TRACK 18, SECTOR 21.
 * 260+66+4=330 WORDS IS ISSUED

* SECTOR 21 SHOULD BE READ CORRECTLY BUT NO MORE
 * READS SHOULD HAPPEN, AGAIN THE 'AOE' BIT SHOULD SET

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

 *TST24: SCOPE

020710	000004			MOV	#STACK, SP	:RESET STACK
020712	012706	001000		MOV	#24, #TSTNM	:SAVE TEST NUMBER
020716	012737	000024	004504			
020724	004737	033004		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
020730	004037	032630		JSR	RO, #FLHEAD	:SAVE HEADER DATA IN WRFROM
020734	002370			WRFROM		:LOCATION WHERE SAVED
020736	000004			4		:NUMBER OF WORDS SAVED
020740	010000			10000		:FIRST DATA WORD
020742	000000			0		:SECOND DATA WORD
020744	000000			0		:THIRD DATA WORD
020746	000000			0		:FOURTH DATA WORD
						:*FILL WRITE FROM BUFFER WITH DATA
020750	004037	032654		JSR	RO, #CLAREA	:CLEAR 256. WORDS, FROM WRFROM+<4*2>
020754	002400			WRFROM+<4*2>		:STARTING FROM WRFROM+<4*2>
020756	000400			256.		:256. WORDS
020760	000000			0		:FILL WITH 0

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

::*****

5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337

```

020762 004037 034756 JSR RO, @#RUN ; SETUP TO RUN FOR DATA COMMAND
020766 000000 0 ; CYLINDER 0
020770 000 ; SECTOR 0
020771 000 ; TRACK 0
020772 177374 -256.-4 ; WORD COUNT (DATA) = 256. +
; 4 HEADER WORDS
020774 002370 WRFROM ; BUS ADDRESS
; STARTING ADDRESS OF DATA
; BUFFER = WRFROM
020776 000000 0 ; DO NOT INHIBIT BUS ADDRESS INCREMENT
021000 010000 FMT22 ; 16 BITS PER WORD FORMAT
; DO NOT INHIBIT ECC CORRECTION
; DO NOT INHIBIT HEADER COMPARE
021002 002344 WRIFOR ; GET READY TO DO A WRIFOR
; WRITE HEADER AND DATA WITH 62 IN RHCS1

021004 004737 033064 JSR PC, @#CHECKT ; CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
; AND THAT ALL STATUS BITS ARE = 0
021010 104400 056662 TYPE ,CPHALT ; CANNOT CONTINUE TESTING IF NOT
021014 000000 HALT ; STOP TEST

021016 013777 004506 161142 MOV @#RPH4VEC, @RPHVEC ; SET RPO4 VECTOR ADDRESS
; TO 'TIME1' IF P-CLOCK IS PRESENT
; OR TO 'TIME2' IF P-CLOCK IS NOT THERE
; 'TIME' WILL ONLY SAVE
; CURRENT CYLINDER ADDRESS
; AND LOOK AHEAD REGISTERS

021024 013746 002344 MOV @#WRIFOR, -(SP) ; GET READY TO MOVE COMMAND
021030 052716 000101 BIS #GO!IE, (SP) ; GET READY TO SET GO AND
; ENABLE INTERRUPT
021034 012677 161140 MOV (SP)+, @RHCS1 ; GO WITH
; 62 IN RHCS1 FOR WRITE HEADER AND DATA
; WITH INTERRUPT ENABLED

021040 104412 WAT ; WAIT FOR RDY BIT TO SET
021042 002200 RHCS1 ; WAIT FOR RHCS1 REGISTER
021044 000200 RDY ; WAIT FOR RDY BIT IN RHCS1 REGISTER
021046 004704 2500. ; ALLOW 2500 MICRO SECONDS
021050 004704 2500. ; RDY MUST SET BETWEEN
; 00 AND 5000 MICRO SECONDS

021052 004737 033004 JSR PC, @#CLDISK ; SET R1-RHCS1, R2-RHCS2
; R3-RHCS1, R4-RHER1
; GIVE RH-11 INITIALIZE
; SETUP UNIT NUMBER

```

5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391

021056 005737 004636
021062 001411

```
;*CHECK THE DRIVE TYPE AND DO THE  
;*APPROPRIATE SECOND WRITE OPERATION  
  
;*FILL WRITE FROM BUFFER WITH HEADER  
:*****  
:*****  
TST      Q#RPO6 ;TEST FOR RPO6 DRIVE  
BEQ      15$      ;TREAT DRIVE AS AN RPO4  
                    ;TREAT AS AN RPO6  
:*****
```

021064 004037 032630
021070 002370
021072 000004
021074 011456
021076 011025
021100 000000
021102 000000
021104 000410

```
JSR      RO,Q#FLHEAD ;SAVE HEADER DATA IN WRFROM  
WRFROM   ;LOCATION WHERE SAVED  
4        ;NUMBER OF WORDS SAVED  
11456    ;FIRST DATA WORD  
<18.*400>!<21.> ;SECOND DATA WORD  
0        ;THIRD DATA WORD  
0        ;FOURTH DATA WORD  
BR       16$ ;CONTINUE WITH THE SECOND WRITE
```

021106

15\$:

021106 004037 032630
021112 002370
021114 000004
021116 010632
021120 011025
021122 000000
021124 000000
021126

```
JSR      RO,Q#FLHEAD ;SAVE HEADER DATA IN WRFROM  
WRFROM   ;LOCATION WHERE SAVED  
4        ;NUMBER OF WORDS SAVED  
10632    ;FIRST DATA WORD  
<18.*400>!<21.> ;SECOND DATA WORD  
0        ;THIRD DATA WORD  
0        ;FOURTH DATA WORD  
16$:    ;CONTINUE WRITE
```

;*FILL WRITE FROM BUFFER WITH DATA - 65125

021126 004037 032654
021132 002400
021134 000400
021136 065125

```
JSR      RO,Q#CLAREA ;CLEAR 256. WORDS. FROM WRFROM+<4*2>  
WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>  
256. ;256. WORDS  
<26.*2000>!<18.*40>!<21.> ;FILL WITH <26.*2000>!<18.*40>!<
```

```
;*CHECK THE DRIVE TYPE AND  
;*FILL WRITE FROM BUFFER WITH APPROPRIATE NEXT HEADER  
  
;*THIS IS A NON EXISTANT HEADER AND SHOULD NOT BE WRITTEN  
;*SINCE 'AOE' SHOULD INHIBIT THE WRITE OPERATION
```

021140 005737 004636

```
:*****  
TST      Q#RPO6 ;TEST FOR RPO6 DRIVE
```



```

5446 021244 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5447 021246 010000 FMT22 ;16 BITS PER WORD FORMAT
5448 ;DO NOT INHIBIT ECC CORRECTION
5449 ;DO NOT INHIBIT HEADER COMPARE
5450 021250 002344 WRIFOR ;GET READY TO DO A WRIFOR
5451 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
5452 021252 000411 BR 8$
5453
5454 021254 7$:
5455
5456 021254 004037 034766 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
5457 021250 000632 410. ;CYLINDER 410.
5458 021252 025 .BYTE 21. ;SECTOR 21.
5459 021263 022 .BYTE 18. ;TRACK 18.
5460 021264 177373 -257.-4 ;WORD COUNT (DATA) = 257. +
5461 ;4 HEADER WORDS
5462 021266 002370 WRFROM ;BUS ADDRESS
5463 ;STARTING ADDRESS OF DATA
5464 ;BUFFER = WRFROM
5465 021270 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5466 021272 010000 FMT22 ;16 BITS PER WORD FORMAT
5467 ;DO NOT INHIBIT ECC CORRECTION
5468 ;DO NOT INHIBIT HEADER COMPARE
5469 021274 002344 WRIFOR ;GET READY TO DO A WRIFOR
5470 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
5471
5472 021276 8$:
5473
5474 ;*SAVE REGISTERS FOR COMPARISON AFTER WIRTE HEADER AND DATA
5475 021276 004037 033152 JSR RO,@#SAVEP ;SAVE REGISTERS
5476 021302 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5477 021304 004512 SAVERE ;STARTING ADDRES OF WHERE
5478 ;THE REGISTERS ARE SAVED
5479 021306 000022 18. ;NUMBER OF REGISTERS
5480 ;SAVED = 19.
5481
5482 021310 004737 033064 JSR PC,@#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
5483 ;AND THAT ALL STATUS BITS ARE = 0
5484 021314 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
5485 021320 000000 HALT ;STOP TEST
5486
5487 021322 013777 004506 160536 MOV @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
5488 ;TO 'TIME1' IF P-CLOCK IS PRESENT
5489 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5490 ;'TIME' WILL ONLY SAVE
5491 ;CURRENT CYLINDER ADDRESS
5492 ;AND LOOK AHEAD REGISTERS
5493
5494
5495 021330 013746 002344 MOV @#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
5496 021334 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
5497 ;ENABLE INTERRUPT
5498 021340 012677 160634 MOV (SP)+, @RHCS1 ;GO WITH
5499 ;62 IN RHCS1 FOR WRITE HEADER AND DATA

```

```

5500 ;WITH INTERRUPT ENABLED
5501
5502
5503 021344 104412 WAT ;WAIT FOR RDY BIT TO SET
5504 021346 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5505 021350 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5506 021352 004704 2500. ;ALLOW 25000 MICRO SECONDS
5507 021354 004704 2500. ;RDY MUST SET BETWEEN
5508 ;00 AND 50000 MICRO SECONDS
5509
5510 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5511
5512 021356 004037 032706 JSR RO, @#FILLRE ;MOV WRFROM+<260.*2>+<1.*2> INTO SAVED RHBA
5513 021362 002174 RHBA ;SAVED REGISTER TO CHANGE
5514 021364 003402 WRFROM+<260.*2>+<1.*2> ;DATA
5515
5516
5517 021366 004037 032706 JSR RO, @#FILLRE ;MOV 0 INTO SAVED PHWC
5518 021372 002172 RHWC ;SAVED REGISTER TO CHANGE
5519 021374 000000 0 ;DATA
5520
5521
5522 021376 004037 033674 JSR RO, @#CHREG ;CHANGE BITS IN SAVED REGISTER
5523 021402 002200 RHCS1 ;CHANGE RHCS1 REGISTER
5524
5525 2 ;2 BIT/BITS TO BE CHANGED
5526 021406 000001 1 ;NEW VALUE OF SC IS 1
5527 021410 100000 SC ;CHANGE SC BIT
5528 021412 000001 1 ;NEW VALUE OF TRE IS 1
5529 021414 040000 TRE ;CHANGE TRE BIT
5530
5531 021416 004037 033674 JSR RO, @#CHREG ;CHANGE BITS IN SAVED REGISTER
5532 021422 002176 RHCS2 ;CHANGE RHCS2 REGISTER
5533
5534 2 ;2 BIT/BITS TO BE CHANGED
5535 021426 000001 1 ;NEW VALUE OF OR IS 1
5536 021430 000200 OR ;CHANGE OR BIT
5537 021432 000001 1 ;NEW VALUE OF IR IS 1
5538 021434 000100 IR ;CHANGE IR BIT
5539
5540 021436 004037 032706 JSR RO, @#FILLRE ;MOV AOE INTO SAVED RHER1
5541 021442 002202 RHER1 ;SAVED REGISTER TO CHANGE
5542 021444 001000 AOE ;DATA
5543
5544
5545 021446 004037 033674 JSR RO, @#CHREG ;CHANGE BITS IN SAVED REGISTER
5546 021452 002222 RHDS1 ;CHANGE RHDS1 REGISTER
5547
5548 3 ;3 BIT/BITS TO BE CHANGED
5549 021456 000001 1 ;NEW VALUE OF ATA IS 1
5550 021460 100000 ATA ;CHANGE ATA BIT
5551 021462 000001 1 ;NEW VALUE OF ERR IS 1
5552 021464 040000 ERR ;CHANGE ERR BIT
5553 021466 000001 1 ;NEW VALUE OF LBT IS 1

```

```

5554 021470 J0200C          LBT          ;CHANGE LBT BIT
5555
5556 021472 053737 004644 004536  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
5557                                ;FOR WORKING DRIVE IN
5558                                ;SAVED RHAS LOCATION
5559
5560                                ;*CHECK DEVICE TYPE BEFORE SETTING UP 'RHCA' & 'RHCC'
5561
5562                                ;:*****
5563 021500 005737 004636  ;:*****
5564 021504 001411          TST      @#RPO6 ;TEST FOR RPO6 DRIVE
5565                                BEQ      9$      ;TREAT AS RPO4
5566                                ;TREAT AS RPO6
5567                                ;:*****
5568
5569 021506 004037 032706  JSR      RO,@#FILLRE ;MOV 815. INTO SAVED RHCA
5570 021512 002212          RHCA          ;SAVED REGISTER TO CHANGE
5571 021514 001457          815.          ;DATA
5572
5573
5574 021516 004037 032706  JSR      RO,@#FILLRE ;MOV 814. INTO SAVED RHCC
5575 021522 002234          RHCC          ;SAVED REGISTER TO CHANGE
5576 021524 001456          814.          ;DATA
5577
5578 021526 000410          BR       10$      ;CONTINUE WITH TEST
5579 021530          9$:
5580
5581 021530 004037 032706  JSR      RO,@#FILLRE ;MOV 411. INTO SAVED RHCA
5582 021534 002212          RHCA          ;SAVED REGISTER TO CHANGE
5583 021536 000633          411.          ;DATA
5584
5585
5586 021540 004037 032706  JSR      RO,@#FILLRE ;MOV 410. INTO SAVED RHCC
5587 021544 002234          RHCC          ;SAVED REGISTER TO CHANGE
5588 021546 000632          410.          ;DATA
5589
5590 021550          10$:
5591                                ;CONTINUE WITH TEST
5592
5593
5594 021550 017737 160430 004524  MOV      @#RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
5595                                ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA WITH AFTER
5596
5597
5598 021556 004037 034002  JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
5599                                ;PRESENT VALUE
5600 021562 004512          SAVERE      ;GOOD DATA SAVED IN 'SAVERE'
5601 021564 002254          WC         ;TEST DATA STARTING FROM 'RHWC'
5602 021566 000022          18.        ;18. REGISTERS TO BE COMPARED
5603 021570 021574          1$         ;RETURN TO 1$ ON ERROR
5604 021572 021600          2$         ;RETURN TO 2$ ON NO ERROR
5605
5606
5607 021574 104055          1$:  ERROR  55      ;WRITING HEADER AND DATA WITH

```

B14

30-MAR-76 18:59 PAGE 121

SEE 0.69

FOR: W. B. G. JOE

...: ADDRESS COVERED BY ...
...: REGISTER ...
...: DATA GIVES ...
...: DAY GIVES ...

...: ...
...: ...
...: ...

021600
021601
021602
021603
021604
021605
021606
021607
021608
021609
021610
021611
021612
021613
021614
021615
021616
021617
021618
021619
021620
021621
021622
021623
021624
021625
021626
021627
021628
021629
021630
021631
021632
021633
021634
021635
021636
021637
021638
021639
021640
021641
021642
021643
021644
021645
021646
021647
021648
021649
021650
021651
021652
021653
021654
021655
021656
021657
021658
021659
021660
021661
021662
021663
021664
021665
021666
021667
021668
021669
021670
021671
021672
021673
021674
021675
021676
021677
021678
021679
021680
021681
021682
021683
021684
021685
021686
021687
021688
021689
021690
021691
021692
021693
021694
021695
021696
021697
021698
021699
021700
021701
021702
021703
021704
021705
021706
021707
021708
021709
021710
021711
021712
021713
021714
021715
021716
021717
021718
021719
021720
021721
021722
021723
021724
021725
021726
021727
021728
021729
021730
021731
021732
021733
021734
021735
021736
021737
021738
021739
021740
021741
021742
021743
021744
021745
021746
021747
021748
021749
021750
021751
021752
021753
021754
021755
021756
021757
021758
021759
021760
021761
021762
021763
021764
021765
021766
021767
021768
021769
021770
021771
021772
021773
021774
021775
021776
021777
021778
021779
021780
021781
021782
021783
021784
021785
021786
021787
021788
021789
021790
021791
021792
021793
021794
021795
021796
021797
021798
021799
021800
021801
021802
021803
021804
021805
021806
021807
021808
021809
021810
021811
021812
021813
021814
021815
021816
021817
021818
021819
021820
021821
021822
021823
021824
021825
021826
021827
021828
021829
021830
021831
021832
021833
021834
021835
021836
021837
021838
021839
021840
021841
021842
021843
021844
021845
021846
021847
021848
021849
021850
021851
021852
021853
021854
021855
021856
021857
021858
021859
021860
021861
021862
021863
021864
021865
021866
021867
021868
021869
021870
021871
021872
021873
021874
021875
021876
021877
021878
021879
021880
021881
021882
021883
021884
021885
021886
021887
021888
021889
021890
021891
021892
021893
021894
021895
021896
021897
021898
021899
021900
021901
021902
021903
021904
021905
021906
021907
021908
021909
021910
021911
021912
021913
021914
021915
021916
021917
021918
021919
021920
021921
021922
021923
021924
021925
021926
021927
021928
021929
021930
021931
021932
021933
021934
021935
021936
021937
021938
021939
021940
021941
021942
021943
021944
021945
021946
021947
021948
021949
021950
021951
021952
021953
021954
021955
021956
021957
021958
021959
021960
021961
021962
021963
021964
021965
021966
021967
021968
021969
021970
021971
021972
021973
021974
021975
021976
021977
021978
021979
021980
021981
021982
021983
021984
021985
021986
021987
021988
021989
021990
021991
021992
021993
021994
021995
021996
021997
021998
021999
022000

;*NOW PREPARE TO DO A READ HEADER AND DATA
;*(THE FIRST READ OPERATION)

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

::*****
2\$:

021600

021600 004737 033034

JSR PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

::*****

021604 005737 004636
021610 001411

TST 0#RPO6 ;TEST FOR RPO6 DRIVE
BEQ 19\$;TREAT UNIT AS AN RPO4
;TREAT AS AN RPO6

::*****

021612 004037 032630
021616 002370
021620 000004
021624 011456
021628 011025
021632 000000
021636 000000
021640 000400

JSR RC,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
11456 ;FIRST DATA WORD
<18.*400>!<21.> ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
BR 20\$;CONTINUE WITH TEST

021634

19\$:

021634 004037 032630
021640 002370
021644 000004
021648 010632
021652 011025
021656 000000
021660 000000
021664

JSR RD,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10632 ;FIRST DATA WORD
<18.*400>!<21.> ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
CONTINUE

20\$:

;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

021654 004037 032654
021660 002400
021664 000400
021668 065125

JSR RD,0#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+(4*2)
WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
256. ;256. WORDS
<26.*2000>!<18.*40>!<21.> ;FILL WITH <26.*2000>!<18.*40>!

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

021666
021672
021674
021676
021700
021702
021704
021706
021712
021716
021722
021724
021730
021732
021734
021736
021740
021742
021744
021746
021750
021750
021754
021756
021757
021760

004037 032654
003400
000006
000377
004037 032654
003434
000412
000377
004737 033004
005737 004636
001412
004037 034766
001456
0225
0222
177266
003434
000000
014000
002350
000411
004037 034766
000332
0225
0222
177266

```
JSR RO, @CLAREA ;CLEAR 6 WORDS FROM WRFROM+(260.*2)  
WRFROM+(260.*2) ;STARTING FROM WRFROM+'260.*2)  
6 ;6 WORDS  
377 ;FILL WITH 377  
  
;*CLEAR READ INTO BUFFER  
  
JSR RC, @CLAREA ;CLEAR 266. WORDS FROM REINTC  
REINTC ;STARTING FROM RE:NTC  
266. ;266. WORDS  
377 ;FILL WITH 377  
  
JSR PC, @CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
  
;*CHECK THE DRIVE TYPE AND DO THE APPROPRIATE  
;*FILL READ HEADER AND DATA COMMAND  
  
:*****  
TST @RPO6 ;TEST FOR RPO6 DRIVE  
BEQ 115 ;TREAT UNIT AS AN RPO4  
;TREAT UNIT AS AN RPO6  
:*****  
  
JSR RO, @RUN ;SETUP TO RUN FOR DATA COMMAND  
814. ;CYLINDER 814.  
.BYTE 21. ;SECTOR 21.  
.BYTE 18. ;TRACK 18.  
-326.-4 ;WORD COUNT (DATA) = 326. +  
4 HEADER WORDS  
REINTC ;BUS ADDRESS  
;STARTING ADDRESS OF DATA  
;BUFFER = REINTC  
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT  
ECI!FMT22 ;16 BITS PER WORD FORMAT  
;INHIBIT ECC CORRECTION  
REFOR ;DO NOT INHIBIT HEADER COMPARE  
;GET READY TO DO A REFOR  
;READ HEADER AND DATA WITH 72 IN RHCS1  
  
BR 125 ;CONTINUE  
  
115:  
  
JSR RO, @RUN ;SETUP TO RUN FOR DATA COMMAND  
410. ;CYLINDER 410.  
.BYTE 21. ;SECTOR 21.  
.BYTE 18. ;TRACK 18.  
-326.-4 ;WORD COUNT (DATA) = 326. +
```

```

5725 021762 003434 REINTO ;4 HEADER WORDS
5726 ;BUS ADDRESS
5727 ;STARTING ADDRESS OF DATA
5728 ;BUFFER = REINTO
5729 021764 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5730 021766 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
5731 ;INHIBIT ECC CORRECTION
5732 021770 002350 REFOR ;DO NOT INHIBIT HEADER COMPARE
5733 ;GET READY TO DO A REFOR
5734 ;READ HEADER AND DATA WITH 72 IN RHCS1
5735
5736 021772 125: ;CONTINUE WITH TESTING
5737
5738 ;*SAVE REGISTERS FOR COMPARISON AFTER
5739 ;*READ HEADER AND DATA
5740 021772 004037 033152 JSR RD,2#SAVER ;SAVE REGISTERS
5741 021776 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5742 022000 004512 SAVERE ;STARTING ADDRESS OF WHERE
5743 ;THE REGISTERS ARE SAVED
5744 ;NUMBER OF REGISTERS
5745 022002 000022 18. ;SAVED = 18.
5746
5747 022004 004737 033064 JSR PC,2#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
5748 ;AND THAT ALL STATUS BITS ARE = 0
5749 022010 104400 056662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
5750 022014 003000 HALT ;STOP TEST
5751
5752 022016 013777 004506 160142 MOV 2#RPO4VEC,2#RPVEC ;SET RPO4 VECTOR ADDRESS
5753 ;TO 'TIME1' IF P-CLOCK IS PRESENT
5754 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5755 ;'TIME' WILL ONLY SAVE
5756 ;CURRENT CYLINDER ADDRESS
5757 ;AND LOOK AHEAD REGISTERS
5758
5759 022024 013746 002350 MOV 2#REFOR,-(SP) ;GET READY TO MOVE COMMAND
5760 022030 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5761 ;ENABLE INTERRUPT
5762 022034 012677 160140 MOV (SP)+,2#RHCS1 ;GO WITH
5763 ;72 IN RHCS1 FOR READ DATA
5764 ;WITH INTERRUPT ENABLED
5765
5766
5767
5768
5769 022040 104412 WAT ;WAIT FOR RDY BIT TO SET
5770 022042 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
5771 022044 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5772 022046 001614 908. ;ALLOW 9080 MICRO SECONDS
5773 022050 001507 839. ;RDY MUST SET BETWEEN
5774 ;690 AND 17470 MICRO SECONDS
5775
5776 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5777
5778 022052 004037 032706 JSR RD,2#FILLRE ;MOV REINTO+(260.*2) INTO SAVED RHBA

```

```

58079 022056 002174          RHBA          ;SAVED REGISTER TO CHANGE
58080 022060 004444          REINT0+(<260.*2) ;DATA
58081
58082
58083 022062 004037 032706    JSR          RO,2#FILLRE ;MOV -70. INTO SAVED RHWC
58084 022066 002172          RHWC          ;SAVED REGISTER TO CHANGE
58085 022070 177672          -70.         ;DATA
58086
58087
58088 022072 004037 033674    JSR          RO,2#CHREG  ;CHANGE BITS IN SAVED REGISTER
58089 022076 002200          RHCS1        ;CHANGE RHCS1 REGISTER
58090
58091 022100 000002          2            ;2 BIT/BITS TO BE CHANGED
58092 022102 000001          1            ;NEW VALUE OF SC IS 1
58093 022104 100000          SC           ;CHANGE SC BIT
58094 022106 000001          1            ;NEW VALUE OF TRE IS 1
58095 022110 040000          TRE         ;CHANGE TRE BIT
58096
58097 022112 004037 033674    JSR          RO,2#CHREG  ;CHANGE BITS IN SAVED REGISTER
58098 022116 002222          RHDS1        ;CHANGE RHDS1 REGISTER
58099
58100 022120 000003          3            ;3 BIT/BITS TO BE CHANGED
58101 022122 000001          1            ;NEW VALUE OF ATA IS 1
58102 022124 100000          ATA         ;CHANGE ATA BIT
58103 022126 000001          1            ;NEW VALUE OF ERR IS 1
58104 022130 040000          ERR         ;CHANGE ERR BIT
58105 022132 000001          1            ;NEW VALUE OF LBT IS 1
58106 022134 002000          LBT         ;CHANGE LBT BIT
58107
58108 022136 004037 032706    JSR          RO,2#FILLRE ;MOV AOE INTO SAVED RHER1
58109 022142 002202          RHER1        ;SAVED REGISTER TO CHANGE
58110 022144 001000          AOE         ;DATA
58111
58112
58113
58114
58115
58116
58117
58118
58119
58120
58121
58122
58123
58124
58125 022164 000404          BR          14$      ;CONTINUE
58126 022166          13$:
58127
58128 022166 004037 032706    JSR          RO,2#FILLRE ;MOV -411. INTO SAVED RHCA
58129 022172 002212          RHCA          ;SAVED REGISTER TO CHANGE
58130 022174 000633          411.         ;DATA
58131
58132 022176          14$:

```

;*CHECK DRIVE TYPE BEFORE SETTING UP 'RHCA'

::*****

```

TST      2#RPO6  ;TEST FOR RPO6 DRIVE
BEG      13$     ;TREAT UNIT AS AN RPO4
          ;TREAT UNIT AS AN RPO6

```

::*****

```

JSR      RO,2#FILLRE ;MOV 815. INTO SAVED RHCA
RHCA     ;SAVED REGISTER TO CHANGE
815.    ;DATA

```

BR 14\$;CONTINUE

13\$:

```

JSR      RO,2#FILLRE ;MOV -411. INTO SAVED RHCA
RHCA     ;SAVED REGISTER TO CHANGE
411.    ;DATA

```

14\$: ;CONTINUE WITH TEST

5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886

```

022176 053737 004644 004536      BIS      @ATTENT,@SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
                                       ;FOR WORKING DRIVE IN
                                       ;SAVED RHAS LOCATION
022204 017737 157774 004524      MOV      @RHDS1,@SAVERE+12 ;RHDS1 IS UNPREDICTABLE
                                       ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
                                       ;*REGISTERS AFTER COMMAND
022212 004037 034002      JSR      RD,@COMREG      ;COMPARE SAVED REGISTERS WITH
                                       ;PRESENT VALUE
022216 004512      SAVERE   ;GOOD DATA SAVED IN 'SAVERE'
022220 002254      WC      ;TEST DATA STARTING FROM 'RHWC'
022222 000022      18.    ;18. REGISTERS TO BE COMPARED
022224 022230      3$    ;RETURN TO 3$ ON ERROR
022226 022234      4$    ;RETURN TO 4$ ON NO ERROR

022230 104055      3$:    ERROR 55      ;READING HEADER AND DATA WITH
022232 000207      RTS      PC      ;EXPECTED ADDRESS OVERFLOW
                                       ;ERROR CAUSED IMPROPER
                                       ;REGISTER CHANGE
                                       ;GOOD DATA GIVES WHAT SHOULD
                                       ;BE THERE
                                       ;RECEIVED DATA GIVES WHAT
                                       ;WAS THERE AFTER COMMAND

                                       ;*NOW COMPARE THE DATA READ
022234      4$:

022234 004037 035032      JSR      RD,@COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
022240 002370      WRFROM  ;GOOD DATA STARTS FROM WRFROM
022242 003424      REINTO  ;TEST DATA STARTS FROM REINTO
022244 000412      266.   ;266. WORDS TO BE COMPARED
022246 022252      5$    ;RETURN TO 5$ ON ERROR
022250 022256      6$    ;RETURN TO 6$ ON NO ERROR

022252 104056      5$:    ERROR 56      ;DATA READ WITH AN EXPECTED
022254 000207      RTS      PC      ;ADDRESS OVERFLOW ERROR
                                       ;IS INCORRECT
                                       ;WORD NO 1 TO 260 SHOULD
                                       ;BE READ CORRECTLY
                                       ;WORD NO 261 TO 266 SHOULD
                                       ;NOT CHANGE DUE TO THE READ

022256      6$:

022256 004737 033004      JSR      PC,@C_DISK    ;SET R1-RHCS1, R2-RHCS2
                                       ;R3-RHDS1, R4-RHER1
                                       ;GIVE RH-11 INITIALIZE
                                       ;SETUP UNIT NUMBER

```

5997
5998
5999
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940

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

::*****

:*FILL WRITE FROM BUFFER WITH EXPECTED HEADER

022262 004037 03263C JSR RC,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

JSR RO,2#CLAREA ;CLEAR 256. WORDS. FROM WRFROM+(4*2)
WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
256. ;256. WORDS
0 ;FILL WITH 0

:*FILL READ INTO BUFFER WITH 377

022314 004037 032654 JSR RC,2#CLAREA ;CLEAR 260. WORDS. FROM REINTC
REINTC ;STARTING FROM REINTC
260. ;260. WORDS
377 ;FILL WITH 377

JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

:*FILL COMMAND FOR READ HEADER AND DATA

022332 004037 034766 JSR RO,2#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256.-4 ;WORD COUNT (DATA) = 256. +
;4 HEADER WORDS
REINTC ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTC
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
ECI!FMT22 ;16 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
REFOR ;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A REFOR
;READ HEADER AND DATA WITH 72 IN RHCS1

```

5941
5942
5943
5944 022354 004037 033152      JSR      RO,#SAVER      ;*SAVE REGISTERS FOR COMPARISON AFTER READ
5945 022360 002172                RHWC      ;SAVE REGISTERS
5946 022362 004512                SAVERE    ;RHWIC IS THE FIRST REGISTER SAVED
5947
5948 022364 000021                17.      ;STARTING ADDRESS OF WHERE
5949
5950 022366 004737 033064      JSR      PC,#CHECKT    ;THE REGISTERS ARE SAVED
5951
5952 022372 104400 056662      TYPE     ;NUMBER OF REGISTERS
5953 022376 000000                HALT     ;SAVED = 17.
5954
5955 022400 013777 004506 157560  MOV     #RP4VEC,#RPVEC ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970 022406 013746 002350      MOV     #REFOR,-(SP)    ;GET READY TO MOVE COMMAND
5971 022412 052716 000101      BIS     #GO!IE,(SP)    ;GET READY TO SET GO AND
5972
5973 022416 012677 157556      MOV     (SP)+,#RHCS1   ;ENABLE INTERRUPT
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985 022422 104412                WAT      ;WAIT FOR RDY BIT TO SET
5986 022424 002200                RHCS1    ;WAIT FOR RHCS1 REGISTER
5987 022426 000200                RDY      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5988 022430 004704                2500.   ;ALLOW 25000 MICRO SECONDS
5989 022432 004704                2500.   ;RDY MUST SET BETWEEN
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500

```

ST22A: ;COMPARE REGISTER BEFORE READ WITH AFTER

```

5995
5996 022464 004037 034002 JSR RD,0#COMREG ;COMPARE SAVED REGISTERS WITH
5997 ;PRESENT VALUE
5998 022470 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5999 022472 002254 WC ;TEST DATA STARTING FROM 'RHWC'
6000 022474 000021 17. ;17. REGISTERS TO BE COMPARED
6001 022476 022502 4$ ;RETURN TO 4$ ON ERROR
6002 022500 022506 1$ ;RETURN TO 1$ ON NO ERROR
6003
6004
6005 022502 104031 4$: ERROR 31 ;READ HEADER AND DATA ON
6006 022504 000207 RTS PC ;CYLINDER 0, SECTOR 0
6007 ;TRACK 0 AFTER A FORCED
6008 ;ADDRESS OVER FLOW ERROR
6009 ;CAUSED IMPROPER REGISTER
6010 ;CHANGE
6011 ;GOOD DATA GIVES WHAT
6012 ;SHOULD BE THERE
6013 ;RECEIVED DATA GIVES WHAT
6014 ;WAS THERE AFTER READ
6015 ;HEADER AND DATA
6016 ;IF HEADER COMPARE ERROR
6017 ;IS FOUND AND THE DATA
6018 ;ERROR GIVES THE NEW
6019 ;HEADER TO
6020 ;CYLINDER 633/1457 (OCTAL)
6021 ;THEN 'AOE' OVER FLOWED
6022 ;INTO HERE
6023
6024 ;*COMPARE DATA READ
6025 022506 1$:
6026
6027 022506 004037 035032 JSR RD,0#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6028 022512 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
6029 022514 003434 REINTO ;TEST DATA STARTS FROM REINTO
6030 022516 000404 260. ;260. WORDS TO BE COMPARED
6031 022520 022524 2$ ;RETURN TO 2$ ON ERROR
6032 022522 022530 3$ ;RETURN TO 3$ ON NO ERROR
6033
6034
6035 022524 104032 2$: ERROR 32 ;READ HEADER AND DATA
6036 022526 000207 RTS PC ;ON CYLINDER 0, TRACK 0
6037 ;SECTOR 0 AFTER A FORCED
6038 ;'AOE' ERROR CAUSED
6039 ;AN ERROR
6040 ;IF FIRST WORD IS
6041 ;10633/11457 (OCTAL) THEN
6042 ;'AOE' OVER FLOWED INTO HERE
6043 022530 3$:
6044
6045
6046

```

```

6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060 022530 000004
6061 022532 012706 001000
6062 022536 012737 000025 004504
6063
6064 022544 004737 033004
6065
6066
6067
6068
6069
6070
6071
6072 022550 004037 032630
6073 022554 002370
6074 022556 000004
6075 022560 010000
6076 022562 000000
6077 022564 000000
6078 022566 000000
6079
6080
6081
6082 022570 004037 032654
6083 022574 002400
6084 022576 000400
6085 022600 000000
6086
6087
6088
6089
6090 022602 004037 034766
6091 022606 000000
6092 022610 000
6093 022611 000
6094 022612 177374
6095
6096 022614 002370
6097
6098
6099 022616 000000
6100 022620 010000

```

```

:*****
:*TEST 25          FORMAT ERROR - RHER1 (BIT #4)FMT
:
:*      AN ATTEMPT WILL BE MADE TO WRITE DATA ON CYLINDER 0
:*      SECTOR 0 TRACK 0 WITH 16 BITS PER WORD WHEN THE
:*      HEADER HAS 16 BITS PER WORD SET
:
:*      THIS SHOULD GIVE FORMAT ERROR FER BIT #4 IN RHER1
:
:*      THEN THIS SECTOR WILL BE READ IN THE CORRECT FORMAT
:*      16 BITS PER WORD TO CHECK THAT NOTHING GOT WRITTEN
:*****
†ST25:  SCOPE
        MOV      #STACK, SP      ;RESET STACK
        MOV      #25, @†STNM     ;SAVE TEST NUMBER
        JSR      PC, @#CLDISK   ;SET R1-RHCS1, R2-RHCS2
                                   ;R3-RHDS1, R4-RHER1
                                   ;GIVE RH-11 INITIALIZE
                                   ;SETUP UNIT NUMBER
:
:*FIRST WRITE HEADER AND DATA CYLINDER 0, TRACK 0, SECTOR 0
:*FILL WRITE FROM BUFFER WITH HEADER
        JSR      RC, @#FLHEAD   ;SAVE HEADER DATA IN WRFROM
        WRFROM  ;LOCATION WHERE SAVED
        4        ;NUMBER OF WORDS SAVED
        10000   ;FIRST DATA WORD
        0        ;SECOND DATA WORD
        0        ;THIRD DATA WORD
        0        ;FOURTH DATA WORD
:
:*FILL WRITE FROM BUFFER WITH DATA
        JSR      RC, @#CLAREA   ;CLEAR 256. WORDS. FROM WRFROM+<4*2>
        WRFROM+<4*2> ;STARTING FROM ~RFROM+<4*2>
        256.    ;256. WORDS
        0        ;FILL WITH 0
:
:*FILL COMMAND
        JSR      RC, @#RUN     ;SETUP TO RUN FOR DATA COMMAND
        0        ;CYLINDER 0
        .BYTE 0 ;SECTOR 0
        .BYTE 0 ;TRACK 0
        -256.-4 ;WORD COUNT (DATA) = 256. +
                                   ;4 HEADER WORDS
        WRFROM  ;BUS ADDRESS
                                   ;STARTING ADDRESS OF DATA
        0        ;BUFFER = WRFROM
                                   ;DO NOT INHIBIT BUS ADDRESS INCREMENT
        FMT22   ;16 BITS PER WORD FORMAT

```

```

6101                                     ;DO NOT INHIBIT ECC CORRECTION
6102                                     ;DO NOT INHIBIT HEADER COMPARE
6103 022622 002344 WRIFOR                ;GET READY TO DO A WRIFOR
6104                                     ;WRITE HEADER AND DATA WITH 62 IN RHCS1
6105
6106
6107 022624 004737 033064 JSR PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND V? = 1
6108                                     ;AND THAT ALL STATUS BITS ARE = 0
6109 022630 104400 056662 TYPE ,CPHALT   ;CANNOT CONTINUE TESTING IF NOT
6110 022634 000000 HALT                  ;STOP TEST
6111
6112 022636 013777 004506 157322 MOV @#RPHVEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
6113                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
6114                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6115                                     ;'TIME' WILL ONLY SAVE
6116                                     ;CURRENT CYLINDER ADDRESS
6117                                     ;AND LOOK AHEAD REGISTERS
6118
6119
6120 022644 013746 002344 MOV @#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
6121 022650 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
6122                                     ;ENABLE INTERRUPT
6123 022654 012677 157320 MOV (SP)+, @RHCS1 ;GO WITH
6124                                     ;62 IN RHCS1 FOR WRITE HEADER AND DATA
6125                                     ;WITH INTERRUPT ENABLED
6126
6127
6128 022660 104412 WAT                      ;WAIT FOR RDY BIT TO SET
6129 022662 002200 RHCS1                 ;WAIT FOR RHCS1 REGISTER
6130 022664 000200 RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6131 022666 004704 2500.                 ;ALLOW 25000 MICRO SECONDS
6132 022670 004704 2500.                 ;RDY MUST SET BETWEEN
6133                                     ;00 AND 50000 MICRO SECONDS
6134
6135
6136 022672 004737 033004 JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
6137                                     ;R3-RHDS1, R4-RHER1
6138                                     ;GIVE RH-11 INITIALIZE
6139                                     ;SETUP UNIT NUMBER
6140

```

ط

```

6141
6142
6143
6144
6145
6146 022676 004037 032654 JSR RD,2*CLAREA ;CLEAR 256. WORDS, FROM WRFROM
6147 022702 002370 WRFROM ;STARTING FROM WRFROM
6148 022704 000400 256. ;256. WORDS
6149 022706 000377 377 ;FILL WITH 377
6150
6151
6152
6153
6154 022710 004037 034766 JSR RD,2*RLN ;SETUP TO RUN FOR DATA COMMAND
6155 022714 000000 0 ;CYLINDER 0
6156 022716 000 .BYTE 0 ;SECTOR 0
6157 022717 000 .BYTE 0 ;TRACK 0
6158 022720 177400 -256. ;WORD COUNT = 256.
6159 022722 002370 WRFROM ;BUS ADDRESS
6160 ;STARTING ADDRESS OF DATA
6161 ;BUFFER = WRFROM
6162 022724 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6163 022726 000000 0 ;18 BITS PER WORD FORMAT
6164 ;DO NOT INHIBIT ECC CORRECTION
6165 ;DO NOT INHIBIT HEADER COMPARE
6166 022730 002342 WRIDAT ;GET READY TO DO A WRIDAT
6167 ;WRITE DATA WITH 60 IN RHCS1
6168
6169
6170
6171 ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
6172 ;*WITH WRONG FORMAT
6173 022732 004037 033152 JSR RD,2*SAVER ;SAVE REGISTERS
6174 022736 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6175 022740 004512 SAVERE ;STARTING ADDRESS OF WHERE
6176 ;THE REGISTERS ARE SAVED
6177 022742 000022 18. ;NUMBER OF REGISTERS
6178 ;SAVED = 18.
6179 022744 004737 033064 JSR PC,3*CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
6180 ;AND THAT ALL STATUS BITS ARE = 0
6181 022750 104400 056662 TYPE .CP=ALT ;CANNOT CONTINUE TESTING IF NOT
6182 022754 000000 HALT ;STOP TEST
6183
6184 022756 013777 004506 157202 MOV 2*RF4VEC,2*RPVEC ;SET RPO4 VECTOR ADDRESS
6185 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6186 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6187 ;'TIME' WILL ONLY SAVE
6188 ;CURRENT CYLINDER ADDRESS
6189 ;AND LOOK AHEAD REGISTERS
6190
6191
6192 022764 013746 002342 MOV 2*WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
6193 022770 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6194 ;ENABLE INTERRUPT

```

```

6195 022774 J12677 157200      MOV      (SP)+, @RHCS1      ;GO WITH
6196                                     ;60 IN RHCS1 FOR WRITE DATA
6197                                     ;WITH INTERRUPT ENABLED
6198
6199
6200 023000 104412      WAT                                     ;WAIT FOR RDY BIT TO SET
6201 023002 002200      RHCS1                                ;WAIT FOR RHCS1 REGISTER
6202 023004 000200      RDY                                  ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6203 023006 001522      650.                                ;ALLOW 8500 MICRO SECONDS
6204 023010 001510      840.                                ;RDY MUST SET BETWEEN
6205                                     ;100 AND 16900 MICRO SECONDS
6206
6207                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
6208 023012 005737 004640      TST      @#RH70                ;RH70 CONTROLLER ?
6209 023016 001411      BEQ      7$                    ;IF NOT, SKIP NEXT RH70 CODE
6210
6211
6212 023020 004037 032706      JSR      RD, @#FILLRE          ;MOV -248. INTO SAVED RHWC
6213 023024 002172      RHWC                                ;SAVED REGISTER TO CHANGE
6214 023026 177410      -248.                             ;DATA
6215
6216
6217 023030 004037 032706      JSR      RD, @#FILLRE          ;MOV WRFROM+<8.*2> INTO SAVED RHBA
6218 023034 002174      RHBA                                ;SAVED REGISTER TO CHANGE
6219 023036 002410      WRFROM+<8.*2>                  ;DATA
6220
6221 023040 000410      BR      8$                    ;SKIP NEXT RH11 CODE
6222
6223
6224
6225 023042                                     7$:
6226
6227 023042 004037 032706      JSR      RD, @#FILLRE          ;MOV -190. INTO SAVED RHWC
6228 023046 002172      RHWC                                ;SAVED REGISTER TO CHANGE
6229 023050 177502      -190.                             ;DATA
6230
6231
6232 023052 004037 032706      JSR      RD, @#FILLRE          ;MOV WRFROM+ <66.*2> INTO SAVED RHBA
6233 023056 002174      RHBA                                ;SAVED REGISTER TO CHANGE
6234 023060 002574      WRFROM+<66.*2>                  ;DATA
6235
6236
6237 023062                                     8$:
6238
6239 023062 004037 033674      JSR      RD, @#CHREG           ;CHANGE BITS IN SAVED REGISTER
6240 023066 002200      RHCS1                            ;CHANGE RHCS1 REGISTER
6241
6242 023070 000002      2                                  ;2 BIT/BITS TO BE CHANGED
6243 023072 000001      1                                  ;NEW VALUE OF SC IS 1
6244 023074 100000      SC                                 ;CHANGE SC BIT
6245 023076 000001      1                                  ;NEW VALUE OF TRE IS 1
6246 023100 040000      TRE                                ;CHANGE TRE BIT
6247
6248 023102 004037 033674      JSR      RD, @#CHREG           ;CHANGE BITS IN SAVED REGISTER
  
```



```

023136 004037 032705
023140 002204
023144 000001
023146 053737 004644 004536
023154 004037 033674
023160 002202
023162 000001
023164 000001
023166 000020
023170 017746 157002
023174 042716 177477
023200 042737 000300 004516
023206 052637 004516
023212 004037 034002
023216 004512
023220 002254
023222 000022
023224 023230
023226 023234
023230 104057
023232 000207

```

```

:MOV 1 INTO SAVED RHCST
:SAVED REGISTER TO CHANGE
:DATA
:CHANGE ERR BIT
:MOV 1 INTO SAVED RHCST
:SAVED REGISTER TO CHANGE
:DATA
:SET APPROPRIATE 'ATA' BITS
:FOR WORKING DRIVE IN
:SAVED RHAS LOCATION
:CHANGE BITS IN SAVED REGISTER
:CHANGE RHER1 REGISTER
:1 BIT/BITS TO BE CHANGED
:NEW VALUE OF FER IS 1
:CHANGE FER BIT
:GET RHCST
:KEEP IR AND OR
:CLEAR SAVED IR OR
:SET OR IR AS REQUIRED
:COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
:COMPARE SAVED REGISTERS WITH
:PRESENT VALUE
:GOOD DATA SAVED IN 'SAVERE'
:TEST DATA STARTING FROM 'RHC'
:18. REGISTERS TO BE COMPARED
:RETURN TO 1$ ON ERROR
:RETURN TO 2$ ON NO ERROR
:ATTEMPTING TO WRITE DATA
:WITH WRONG FORMAT BIT CAUSED
:IMPROPER REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD
:BE THERE
:RECEIVED DATA GIVES WHAT WAS

```

;THERE AFTER ATTEMPTED WRITE

;*NOW PREPARE TO READ WITH CORRECT FORMAT TO CHECK
;*THAT NOTHING GOT WRITTEN

023234

25:

023234 004737 033004

JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

023240 004037 032654

023244 002370
023246 000400
023250 000000

JSR RD,2#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
WRFROM ;STARTING FROM WRFROM
256. ;256. WORDS
0 ;FILL WITH 0

;*FILL READ INTO BUFFER WITH 125252

023252 004037 032654

023256 003434
023260 000400
023262 125252

JSR RD,2#CLAREA ;CLEAR 256. WORDS, FROM REINTO
REINTO ;STARTING FROM REINTO
256. ;256. WORDS
125252 ;FILL WITH 125252

;*FILL COMMAND TO READ DATA

023264 004037 034766

023270 000000
023272 000
023274 000
023276 177400
023278 003434

JSR RD,2#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256. ;WORD COUNT = 256.
REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO
;DO NOT INHIBIT BUS ADDRESS INCREMENT
;16 BITS PER WORD FORMAT

023300 000000
023302 014000

0 ;INHIBIT ECC CORRECTION
FMT22!ECI ;DO NOT INHIBIT HEADER COMPARE

023304 002346

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

023306 004037 033152

023312 002172
023314 004512

;*SAVE REGISTERS FOR COMPARISON AFTER NORMAL READ
JSR RD,2#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVER
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
18. ;NUMBER OF REGISTERS
;SAVED = 18.

023316 000022

023320 004737 033064

JSR PC,2#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0

g

```

023324 104400 056662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
023330 000000 HALT ;STOP TEST

023332 013777 004506 156625 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS

023340 013746 002346 MOV @#READAT,-(SP) ;GET READY TO MOVE COMMAND
023344 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT

023350 012677 156624 MOV (SP)+,@RHCS1 ;GO WITH
;70 IN RHCS1 FOR READ DATA
;WITH INTERRUPT ENABLED

023354 104412 WAT ;WAIT FOR RDY BIT TO SET
023356 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
023360 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
023362 001614 908. ;ALLOW 9080 MICRO SECONDS
023364 001507 839. ;RDY MUST SET BETWEEN
;690 AND 17470 MICRO SECONDS

;*CHANGE SAVED REGISTERS TO EXPECTED VALUE

023366 004037 032706 JSR RD,@#FILLRE ;MOV REINT0+(256.*2) INTO SAVED RHBA
023372 002174 RHBA ;SAVED REGISTER TO CHANGE
023374 004434 REINT0+(256.*2) ;DATA

023376 004037 032706 JSR RD,@#FILLRE ;MOV 0 INTO SAVED RHWC
023402 002172 RHWC ;SAVED REGISTER TO CHANGE
023404 000000 0 ;DATA

023406 004037 032706 JSR RD,@#FILLRE ;MOV 1 INTO SAVED RHDST
023412 002204 RHDST ;SAVED REGISTER TO CHANGE
023414 000001 1 ;DATA

023416 017746 156554 MOV @RHCS2,-(SP) ;GET RHCS2
023422 042716 177477 BIC #!C,IR!OR,(SP) ;KEEP IR AND OR
023426 042737 000300 BIC #IR!OR,@#SAVERE+4 ;CLEAR SAVED IR OR
023434 052637 004516 BIS (SP)+,@#SAVERE+4 ;SET OR IR AS REQUIRED

;*COMPARE REGISTERS BEFORE READ WITH AFTER

023440 004037 034002 JSR RD,@#COMREG ;COMPARE SAVED REGISTERS WITH
;PRESENT VALUE

```

E15

MNDEC-11-DZRIJ-A, RPO4
DZRIJA.P11

5 6 FUNCT. CONT. TST-PT 2
FORMAT ERROR - RHER1 (BIT #4)FMT

MACY11 27(655) 30-MAR-76 18:59 PAGE 137

SEQ C185

023444	004512		SAVERE		:GOOD DATA SAVED IN 'SAVERE'
023446	002254		NC		:TEST DATA STARTING FROM 'PHWC'
023450	000022		19.		:18. REGISTERS TO BE COMPARED
023452	023456		3\$:RETURN TO 3\$ ON ERROR
023454	023462		4\$:RETURN TO 4\$ ON NC ERROR
023456	104033				
023460	000207	3\$:	ERROR	33	:READ DATA AFTER AN
			RTS	PC	:ATTEMPTED WRITE WITH WRONG
					:IMPROPER REGISTER CHANGE
					:FORMAT CAUSED
					:GOOD DATA GIVES WHAT SHOULD
					:BE THERE
					:RECEIVED DATA GIVES WHAT
					:WAS THERE AFTER READ
					:*COMPARE DATA READ AFTER ATTEMPTED WRITE WITH
					:*WRONG FORMAT BIT
023462		4\$:			
023462	004037		JSR	RD.2#COMPAR	:COMPARE TWO BLOCKS OF MEMORY
023466	002370	025032	WRFROM		:GOOD DATA STARTS FROM WRFROM
023470	003434		REINTC		:TEST DATA STARTS FROM REINTC
023472	000400		256.		:256. WORDS TO BE COMPARED
023474	023500		5\$:RETURN TO 5\$ ON ERROR
023476	023504		6\$:RETURN TO 6\$ ON NO ERROR
023500	104034				
023502	000207	5\$:	ERROR	34	:DATA READ AFTER AN ATTEMPT
			RTS	PC	:TO WRITE WITH WRONG FORMAT
					:WAS INCORRECT
023504		6\$:			

F15

6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498

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

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

::*****

†ST26: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #26,3*†STNM ;SAVE TEST NUMBER
JSR PC,3*CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

;*FILL WRITE FROM BUFFER WITH 107070

023524 004037 032654 JSR R0,3*CLAREA ;CLEAR 256. WORDS, FROM WRFROM
023530 002370 WRFROM ;STARTING FROM WRFROM
023532 000400 256. ;256. WORDS
023534 107070 107070 ;FILL WITH 107070

;*FILL READ INTO BUJFFER WITH 107070

023536 004037 032654 JSR R0,3*CLAREA ;CLEAR 256. WORDS, FROM REINTO
023542 003434 REINTO ;STARTING FROM REINTO
023544 000400 256. ;256. WORDS
023546 107070 107070 ;FILL WITH 107070

;*FILL COMMAND TO READ WITH WRONG FORMAT.

023550 004037 034766 JSR R0,3*RUN ;SETUP TO RUN FOR DATA COMMAND
023554 000000 0 ;CYLINDER 0
023556 000 ;SECTOR 0
023557 000 ;TRACK 0
023560 177400 -256. ;WORD COUNT = 256.
023562 003434 REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO

023564 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023566 004000 ECI ;18 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
023570 002346 READAT ;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A READAT
;READ DATA WITH 70 IN RHCS1

;*SAVE REGISTERS FOR COMPARRISON AFTER READ

```

6499 023572 004037 033152 JSR RO,2#SAVER ;SAVE REGISTERS
6500 023576 002172 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6501 023600 004512 SAVERE ;STARTING ADDRESS OF WHERE
6502 ;THE REGISTERS ARE SAVED
6503 023602 000022 18. ;NUMBER OF REGISTERS
6504 ;SAVED = 18.
6505
6506 023604 004737 033064 JSR PC,2#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
6507 ;AND THAT ALL STATUS BITS ARE = 0
6508 023610 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
6509 023614 000000 HALT ;STOP TEST
6510
6511 023616 013777 004506 156342 MOV 2#RPO4VEC,2#RPO4VEC ;SET RPO4 VECTOR ADDRESS
6512 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6513 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6514 ;'TIME' WILL ONLY SAVE
6515 ;CURRENT CYLINDER ADDRESS
6516 ;AND LOOK AHEAD REGISTERS
6517
6518
6519 023624 013746 002346 MOV 2#READAT, -(SP) ;GET READY TO MOVE COMMAND
6520 023630 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
6521 ;ENABLE INTERRUPT
6522 023634 012677 156340 MOV (SP)+,2#RHCS1 ;GO WITH
6523 ;70 IN RHCS1 FOR READ DATA
6524 ;WITH INTERRUPT ENABLED
6525
6526
6527 023640 104412 WAT ;WAIT FOR RDY BIT TO SET
6528 023642 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
6529 023644 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6530 023646 001522 850. ;ALLOW 8500 MICRO SECONDS
6531 023650 001510 840. ;RDY MUST SET BETWEEN
6532 ;100 AND 16900 MICRO SECONDS
6533
6534 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
6535
6536
6537 023652 004037 033674 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
6538 023656 002200 RHCS1 ;CHANGE RHCS1 REGISTER
6539
6540 2 ;2 BIT/BITS TO BE CHANGED
6541 1 ;NEW VALUE OF SC IS 1
6542 023662 000001 SC ;CHANGE SC BIT
6543 1 ;NEW VALUE OF TRE IS 1
6544 023664 100000 TRE ;CHANGE TRE BIT
6545 023666 000001
6546 023670 040000
6547
6548 023672 004037 033674 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
6549 023676 002222 RHCS1 ;CHANGE RHCS1 REGISTER
6550
6551 2 ;2 BIT/BITS TO BE CHANGED
6552 1 ;NEW VALUE OF ATA IS 1
6553 023700 000002 ATA ;CHANGE ATA BIT
6554 023702 000001
6555 023704 100000
6556 023706 000001 1 ;NEW VALUE OF ERR IS 1
    
```

H15

```

6553 023710 040000          ERR          ;CHANGE ERR BIT
6554
6555 023712 004037 032706    JSR      RO,@#FILLRE  ;MOV 1 INTO SAVED RHDST
6556 023716 002204          RHDST     ;SAVED REGISTER TO CHANGE
6557 023720 000001          1         ;DATA
6558
6559
6560 023722 004037 032674    JSR      RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
6561 023726 002202          RHER1     ;CHANGE RHER1 REGISTER
6562
6563
6564 023730 000001          1         ;1 BIT/BITS TO BE CHANGED
6565 023732 000001          1         ;NEW VALUE OF FER IS 1
6566 023734 000020          FER      ;CHANGE FER BIT
6567
6568 023736 053737 004644 004536  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
6569                                     ;FOR WORKING DRIVE IN
6570                                     ;SAVED RHAS LOCATION
6571 023744 017746 156226    MOV      @RHCS2 -(SP)  ;GET RHCS2
6572 023750 042716 177477    BIC     #1<IR!OR> (SP) ;KEEP IR AND OR
6573 023754 042737 000300 004516  BIC     #IR!OR,@#SAVERE+4 ;CLEAR SAVED IR OR
6574 023762 052637 004516    BIS      (SP)+,@#SAVERE+4 ;SET OR IR AS REQUIRED
6575
6576
6577                                     ;*COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
6578
6579 023766 004037 034002    JSR      RO,@#COMREG  ;COMPARE SAVED REGISTERS WITH
6580                                     ;PRESENT VALUE
6581 023772 004512          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
6582 023774 002254          WC      ;TEST DATA STARTING FROM 'RHWC'
6583 023776 000022          19.     ;18. REGISTERS TO BE COMPARED
6584 024000 024004          1$     ;RETURN TO 1$ ON ERROR
6585 024002 024010          2$     ;RETURN TO 2$ ON NO ERROR
6586
6587
6588 024004 104057          1$:     ERROR 57    ;ATTEMPTING TO READ DATA
6589 024006 000207          RTS      PC      ;WITH WRONG FORMAT BIT CAUSED
6590                                     ;IMPROPER REGISTER CHANGE
6591                                     ;GOOD DATA GIVES WHAT SHOULD BE
6592                                     ;THERE
6593                                     ;RECEIVED DATA GIVES WHAT WAS THERE
6594                                     ;AFTER READ DATA
6595
6596                                     ;*COMPARE READ INTO BUFFER TO CHECK THAT NOTHING WAS READ
6597 024010          2$:
6598
6599 024010 004037 035032    JSR      RO,@#COMPAR  ;COMPARE TWO BLOCKS OF MEMORY
6600 024014 002370          WRFROM   ;GOOD DATA STARTS FROM WRFROM
6601 024016 003434          REINTO   ;TEST DATA STARTS FROM REINTO
6602 024020 000400          256.    ;256. WORDS TO BE COMPARED
6603 024022 024026          3$     ;RETURN TO 3$ ON ERROR
6604 024024 024032          4$     ;RETURN TO 4$ ON NO ERROR
6605
6606
    
```

6607	024026	104034	3\$:	ERROR	34
6608	024030	000207		RTS	PC
6609					
6610					
6611					
6612					
6613					
6614					
6615	024032		4\$:		

```

:ATTEMPT TO READ
:WITH WRONG FORMAT BIT
:CHANGED READ INTO BUFFER
:GOOD DATA GIVES WHAT SHOULD
:BE THERE
:BAD DATA GIVES WHAT WAS
:THERE AFTER READ DATA

```




```

6617
6618
6619
6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637 024032 000004
6638 024034 012706 001000
6639 024040 012737 000027 004504
6640
6641 024046 004737 033004
6642
6643
6644
6645 024052 012737 002200 004650
6646 024060 012737 000007 004656
6647
6648
6649
6650
6651 024066
6652
6653 024066 004737 033004
6654
6655
6656
6657
6658 024072 004037 032630
6659 024076 002370
6660 024100 000004
6661 024102 010001
6662 024104 000000
6663 024106 000000
6664 024110 000000
6665
6666
6667
6668 024112 004037 032654
6669 024116 002400
6670 024120 000400

```

```

*****
*TEST 27 REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR
*
* CYLINDER1 TRACK 0, SECTOR 0 WILL BE WRITTEN WITH
* 200 WORDS OF 2000 BY A WRITE HEADER AND DATA COMMAND
*
* THE HEADS WILL BE BROUGHT TO CYLINDER 0 BY A SEEK
*
* A READ DATA COMMAND WILL BE GIVEN TO CYLINDER 1 TRACK 0
* SECTOR 0 150. WORDS. THIS WILL TAKE AT
* LEAST 7 MILI SECONDS. IMMEDIATELY AFTER GO AT
* IMPLIED SEEK TIME, WRITE INTO A REGISTER WILL BE ATTEMPTED
* THEN READY WILL BE WAITED ON TO COMPLETE THE READ DATA
* THEN ALL REGISTERS WILL BE COMPARED AND THE DATA READ
* SHOULD BE GOOD
* THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST, RHER2
* RHOF, RHCA, RHER3
*****
†ST27: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #27,‡†STNM ;SAVE TEST NUMBER
JSR PC,‡‡CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
MOV #RHCS1,‡‡TMPO ;FIRST REGISTER TO BE TESTED
MOV #7,‡‡TMPS ;NUMBER OF REGISTERS TO BE TESTED
; *PREPARE TO WRITE HEADER AND DATA CYLINDER 1, TRACK 0, SECTOR 0
; *FILL WRITE FROM BUFFER WITH HEADER
ST22:
JSR PC,‡‡CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR RO,‡‡FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10001 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
; *FILL WRITE FROM BUFFER WITH DATA
JSR RO,‡‡CLAREA ;CLEAR 256. WORDS, FROM WRFROM+⟨4*2⟩
WRFROM+⟨4*2⟩ ;STARTING FROM WRFROM+⟨4*2⟩
256. ;256. WORDS

```


L15

MNDEC-11-DZRJJ-A, RFO4/5, 6 FUNCT. CONT. TST-PT 2 MACY11 27(655) 30-MAR-76 18:59 PAGE 144
 DZRJJA.P11 †27 REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR

SEQ 0192

6725	024220	104400	056662	TYPE	.CPHALT	:CANNOT CONTINUE TESTING IF NOT
6726	024224	000000		HALT		:STOP TEST
6727						
6728						
6729	024226	004037	032754	JSR	RO, @#SEEKCY	:SEEK FOR
6730	024232	000000		0		:CYLINDER 0
6731						
6732						
6733	024234	013777	004506 155724	MOV	@#RP4VEC, @RPVEC	:SET RPO4 VECTOR ADDRESS
6734						:TO 'TIME1' IF P-CLOCK IS PRESENT
6735						:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6736						: 'TIME' WILL ONLY SAVE
6737						:CURRENT CYLINDER ADDRESS
6738						:AND LOOK AHEAD REGISTERS
6739						
6740						
6741	024242	013746	002352	MOV	@#SEECOM, -(SP)	:GET READY TO MOVE COMMAND
6742	024246	052716	000101	BIS	#GO!IE, (SP)	:GET READY TO SET GO AND
6743						:ENABLE INTERRUPT
6744	024252	012677	155722	MOV	(SP)+, @RHCSI	:GO WITH
6745						:4 IN RHCSI FOR SEEK
6746						:WITH INTERRUPT ENABLED
6747						
6748						
6749	024256	104412		WAT		:WAIT FOR DRY BIT TO SET
6750	024260	002222		RHDS1		:WAIT FOR RHDS1 REGISTER
6751	024262	000200		DRY		:WAIT FOR DRY BIT IN RHDS1 REGISTER
6752	024264	002776		1534.		:ALLOW 15340 MICRO SECONDS
6753	024266	001502		834.		:DRY MUST SET BETWEEN
6754						:7000 AND 23680 MICRO SECONDS
6755						
6756						:*PREPARE FOR A READ DATA
6757						
6758						:*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
6759						
6760	024270	004037	032654	JSR	RO, @#CLAREA	:CLEAR 150. WORDS, FROM WRFROM
6761	024274	002370		WRFROM		:STARTING FROM WRFROM
6762	024276	000226		150.		:150. WORDS
6763	024300	002000		2000		:FILL WITH 2000
6764						
6765						
6766	024302	004037	032654	JSR	RO, @#CLAREA	:CLEAR 106. WORDS, FROM WRFROM+(150.*2)
6767	024306	003044		WRFROM+(150.*2)		:STARTING FROM WRFROM+(150.*2)
6768	024310	000152		106.		:106. WORDS
6769	024312	000077		77		:FILL WITH 77
6770						
6771						
6772						:*FILL READ INTO BUFFER WITH DATA OTHER THAN WHAT IS EXPECTED
6773						
6774	024314	004037	032654	JSR	RO, @#CLAREA	:CLEAR 256. WORDS, FROM REINTO
6775	024320	003434		REINTO		:STARTING FROM REINTO
6776	024322	000400		256.		:256. WORDS
6777	024324	000077		77		:FILL WITH 77
6778						

M15

```

6779                ;*FILL READ DATA COMMAND
6780
6781 024326 004037 034766 JSR      RO,@#RUN      ;SETUP TO RUN FOR DATA COMMAND
6782 024332 000001          1          ;CYLINDER 1
6783 024334          000          ;SECTOR 0
6784 024335          000          ;TRACK 0
6785 024336 177552          ;WORD COUNT = 150.
6786 024340 003434          ;BUS ADDRESS
6787                ;STARTING ADDRESS OF DATA
6788                ;BUFFER = REINTO
6789 024342 000000          0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6790 024344 014000          ECI!FMT22 ;16 BITS PER WORD FORMAT
6791                ;INHIBIT ECC CORRECTION
6792                ;DO NOT INHIBIT HEADER COMPARE
6793 024346 002346          READAT    ;GET READY TO DO A READAT
6794                ;READ DATA WITH 70 IN RHCS1
6795
6796
6797                ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE
6798                ;*INTO A REGISTER WHILE THE READ IS GOING ON
6799 024350 004037 033152 JSR      RO,@#SAVER    ;SAVE REGISTERS
6800 024354 002172          RHWC      ;RHWC IS THE FIRST REGISTER SAVED
6801 024356 004512          SAVERE    ;STARTING ADDRESS OF WHERE
6802                ;THE REGISTERS ARE SAVED
6803 024360 000022          18.        ;NUMBER OF REGISTERS
6804                ;SAVED = 19.
6805
6806 024362 004737 033064 JSR      PC,@#CHECKT   ;CHECKS DVA, RDY, MCL, DPR, DRY AND VV = 1
6807                ;AND THAT ALL STATUS BITS ARE = 0
6808 024366 104400 056662 TYPE      ,CPHALT      ;CANNOT CONTINUE TESTING IF NOT
6809 024372 000000          HALT      ;STOP TEST
6810
6811 024374 013777 004506 155564 MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
6812                ;TO 'TIME1' IF P-CLOCK IS PRESENT
6813                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6814                ;'TIME' WILL ONLY SAVE
6815                ;CURRENT CYLINDER ADDRESS
6816                ;AND LOOK AHEAD REGISTERS
6817
6818
6819 024402 013746 002346 MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
6820 024406 052716 000101 BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
6821                ;ENABLE INTERRUPT
6822 024412 012677 155562 MOV      (SP)+,@RHCS1   ;GO WITH
6823                ;70 IN RHCS1 FOR READ DATA
6824                ;WITH INTERRUPT ENABLED
6825
6826 024416 013700 004650 MOV      @#TMPO,RO      ;SET UP RO FOR WRITE
6827 024422 012737 002006 MOV      #BIT1!BIT2!BIT10,@(RO)+ ;ATTEMPT TO WRITE INTO
6828                ;REGISTERS DURING IMPLIED SEE:
6829 024426 010037 004650 MOV      RO,@#TMPO     ;SAVE OFF RO
6830
6831                ;*NOW RMR IS SET BUT THE COMPLETION OF READ MUST BE
6832                ;*WAITED ON

```

N15

MNDEC-11-DZRJJ-A, RPO4.5/6 FUNCT. CONT. TST-PT 2 MACY11 27 (655) 30-MAR-76 18:59 PAGE 146
 DZRJJA.P11 127 REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR

SEG 0194

```

6833
6834
6835 024432 104412      WAT      ;WAIT FOR RDY BIT TO SET
6836 024434 002200      RHCS1   ;WAIT FOR RHCS1 REGISTER
6837 024436 000200      RDY     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6838 024440 002237      1183.  ;ALLOW 11930 MICRO SECONDS
6839 024442 002237      1183.  ;RDY MUST SET BETWEEN
6840                                     ;00 AND 23660 MICRO SECONDS
6841
6842                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
6843
6844 024444 004037 033674  JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
6845 024450 002234      RHCC    ;CHANGE RHCC REGISTER
6846
6847 024452 000001      1       ;1 BIT/BITS TO BE CHANGED
6848 024454 000001      1       ;NEW VALUE OF BIT0 IS 1
6849 024456 000001      BIT0    ;CHANGE BIT0 BIT
6850
6851 024460 004037 033674  JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
6852 024464 002200      RHCS1   ;CHANGE RHCS1 REGISTER
6853
6854 024466 000001      1       ;1 BIT/BITS TO BE CHANGED
6855 024470 000001      1       ;NEW VALUE OF SC IS 1
6856 024472 000000      SC      ;CHANGE SC BIT
6857
6858 024474 004037 032706  JSR      RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
6859 024500 002204      RHDST  ;SAVED REGISTER TO CHANGE
6860 024502 000001      1       ;DATA
6861
6862
6863 024504 004037 033674  JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
6864 024510 002222      RHCS1   ;CHANGE RHDS1 REGISTER
6865
6866 024512 000002      2       ;2 BIT/BITS TO BE CHANGED
6867 024514 000001      1       ;NEW VALUE OF ATA IS 1
6868 024516 100000      ATA    ;CHANGE ATA BIT
6869 024520 000001      1       ;NEW VALUE OF ERR IS 1
6870 024522 040000      ERR    ;CHANGE ERR BIT
6871
6872 024524 053737 004644 004536  BIS      @#ATTENT,@#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
6873                                     ;FOR WORKING DRIVE IN
6874                                     ;SAVED RHAS LOCATION
6875
6876 024532 004037 033674  JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
6877 024536 002202      RHER1   ;CHANGE RHER1 REGISTER
6878
6879 024540 000001      1       ;1 BIT/BITS TO BE CHANGED
6880 024542 000001      1       ;NEW VALUE OF RMR IS 1
6881 024544 000004      RMR    ;CHANGE RMR BIT
6882
6883 024546 004037 032706  JSR      RO,@#FILLRE ;MOV REINTO+<150.*2> INTO SAVED RHBA
6884 024552 002174      RHBA   ;SAVED REGISTER TO CHANGE
6885 024554 004110      REINTO+<150.*2> ;DATA
6886
  
```

024500	004037	032706	JSR	RO, 2#FILLRE	:MOV 0 INTO SAVED RHWC
024504	004037		RHWC		:SAVED REGISTER TO CHANGE
024508	004037		0		:DATA
:*COMPARE REGISTERS BEFORE READ DATA WITH REGISTERS					
:*AFTER READ AND ATTEMPTED MODIFICATION OF REGISTER					
02450C	004037	034002	JSR	RO, 2#COMREG	:COMPARE SAVED REGISTERS WITH
024510	004512		SAVERE		:PRESENT VALUE
024514	002224		NO		:GOOD DATA SAVED IN 'SAVERE'
024518	000022		18.		:TEST DATA STARTING FROM 'RHWC'
02451C	024604		25.		:18. REGISTERS TO BE COMPARED
024520	024624		35		:RETURN TO 25 ON ERROR
024524	024624				:RETURN TO 35 ON NO ERROR
024528	010046	25:	MOV	RO, -(SP)	:PUSH RO ON STACK
024530	013700		MOV	2#TMP1, RO	:GET REGISTER BEING MODIFIED + 2 POINTER
024534	014237	004650	MOV	-(RO), 2#SBDADR	:GET ADDRESS OF REGISTER BEING MODIFIED
024538	104060	001122	ERROR	6C	:ATTEMPTING TO MODIFY REGISTER
02453C	012500		MOV	(SP)+, RO	:POP STACK INTO RO
024540	000207		RTS	PC	:DURING A READ COMMAND CAUSED
:IMPROPER REGISTER CHANGE					
:GOOD DATA GIVES WHAT SHOULD					
:BE THERE					
:RECEIVE DATA GIVES WHAT WAS					
:THERE AFTER READ					
:*COMPARE DATA READ					
024544		35:			
024548	004037	035032	JSR	RO, 2#COMPAR	:COMPARE TWO BLOCKS OF MEMORY
02454C	002224		WAFROM		:GOOD DATA STARTS FROM WAFROM
024550	000022		REINTO		:TEST DATA STARTS FROM REINTO
024554	004674		45		:45. WORDS TO BE COMPARED
024558	004674		ST23		:RETURN TO ST23 ON ERROR
02455C					:RETURN TO 35 ON NO ERROR
024560	104034	45:	ERROR	34	:DATA READ WITH AN ATTEMPTED
024564	000207		RTS	PC	:MODIFICATION OF REGISTER
:DURING READ CAUSED ERROR					
024568	005327	004656	ST23:	DEC	2#TMP5
02456C	000002		BNE	15	:COUNT DOWN
024570	000137	024652	JMP	TST30	:BRANCH IF 7 NOT DONE
024574	000137	024056	JMP	2#ST22	:JUMP TO NEXT TEST
:JUMP TO BEGINNING OF TEST					

69.90
69.91
69.92
69.93
69.94
69.95
69.96
69.97
69.98
69.99
70.00
70.01
70.02
70.03
70.04
70.05
70.06
70.07
70.08
70.09
70.10
70.11
70.12
70.13
70.14
70.15
70.16
70.17
70.18
70.19
70.20
70.21
70.22
70.23
70.24
70.25
70.26
70.27
70.28
70.29
70.30
70.31
70.32
70.33
70.34
70.35
70.36
70.37
70.38
70.39
70.40
70.41
70.42

```

;*FILL WRITE FROM BUFFER WITH DATA
024742 004037 032654 JSR RO,0#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+(4*2)
024746 002400 WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
024750 000400 256. ;256. WORDS
024752 070707 ;FILL WITH 070707

;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
024754 004037 032630 JSR RO,0#FLHEAD ;SAVE HEADER DATA IN WRFROM+(260.*2)
024760 003400 WRFROM+(260.*2) ;LOCATION WHERE SAVED
024762 000004 4 ;NUMBER OF WORDS SAVED
024764 010001 10001 ;FIRST DATA WORD
024766 000001 1 ;SECOND DATA WORD
024770 000000 0 ;THIRD DATA WORD
024772 000000 0 ;FOURTH DATA WORD

;*FILL WRITE FROM BUFFER WITH WITH NEXT SECTOR DATA
024754 004037 032654 JSR RO,0#CLAREA ;CLEAR 4 WORDS, FROM WRFROM+(248.*2)
025000 003420 WRFROM+(260.*2) ;STARTING FROM WRFROM+(260.*2)
025002 000004 4 ;4 WORDS
025004 070707 ;FILL WITH 70707

;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
025006 004037 034766 JSR RO,0#RUN ;SETUP TO RUN FOR DATA COMMAND
025012 000001 1 ;CYLINDER 1
025014 000 ;SECTOR 0
025015 000 ;TRACK 0
025016 177364 -264.-4 ;WORD COUNT (DATA) = 264. +
;4 HEADER WORDS
WRFROM ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
025022 000000 0 FMT22 ;16 BITS PER WORD FORMAT
025024 010000 ;DO NOT INHIBIT ECC CORRECTION
025026 002344 WRIFOR ;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A WRIFOR
;WRITE HEADER AND DATA WITH 62 IN RHCSI

025030 004737 033064 JSR PC,0#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
025034 104400 056662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
025040 000000 HALT ;STOP TEST

025042 013777 004506 155116 MO. 0#RP4VEC,0RPVEC ;SET RP04 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE

```


E16

7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097

025050 013746 002344
025054 052716 000101
025060 012677 155114

025064 104412
025066 002200
025070 000200
025072 001725
025074 001502

025076 004037 032654
025102 002370
025104 000400
025106 002000

025110 004037 032654
025114 003370
025116 000004
025120 002001

025122 004037 034766
025126 000001
025130 000
025131 000
025132 177400
025134 002370

025136 000000
025140 010000

025142 002342

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

MOV 0*WRIFOR, -(SP) : GET READY TO MOVE COMMAND
SIS 0*GO! IE, (SP) : GET READY TO SET GO AND
: ENABLE INTERRUPT
MOV (SP)+, 0RHCS1 : GO WITH
: 62 IN RHCS1 FOR WRITE HEADER AND DATA
: WITH INTERRUPT ENABLED

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

WAT : WAIT FOR RDY BIT TO SET
RHCS1 : WAIT FOR RHCS1 REGISTER
RDY : WAIT FOR RDY BIT IN RHCS1 REGISTER
981. : ALLOW 9810 MICRO SECONDS
834. : RDY MUST SET BETWEEN
: 1470 AND 18150 MICRO SECONDS

: *NOW PREPARE FOR THE WRITE DATA COMMAND
: *FILL WRITE FROM BUFFER WITH 256 OF 2000 AND 4 OF 2001

JSR 00, 0*CLAREA : CLEAR 256. WORDS, FROM WRFROM
WRFROM : STARTING FROM WRFROM
256. : 256. WORDS
2000 : FILL WITH 2000

JSR 00, 0*CLAREA : CLEAR 4 WORDS, FROM WRFROM+(256.*2)
WRFROM+(256.*2) : STARTING FROM WRFROM+(256.*2)
4 : 4 WORDS
2001 : FILL WITH 2001

: *FILL WRITE DATA COMMAND

JSR 00, 0*FJN : SETUP TO RUN FOR DATA COMMAND
1 : CYLINDER 1
.BYTE 0 : SECTOR 0
.BYTE 0 : TRACK 0
-256. : WORD COUNT = 256.
WRFROM : BUS ADDRESS
: STARTING ADDRESS OF DATA
: BUFFER = WRFROM
0 : DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 : 16 BITS PER WORD FORMAT
: DO NOT INHIBIT ECC CORRECTION
: DO NOT INHIBIT HEADER COMPARE
WRIDAT : GET READY TO DO A WRIDAT
: WRITE DATA WITH 60 IN RHCS1

F16

```

7100  : *SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED
7101  : *REGISTER MODIFICATION DURING A WRITE DATA
7102  JSR      RO, @SAVER      :SAVE REGISTERS
7103  RHWC      :RHWC IS THE FIRST REGISTER SAVED
7104  SAVERE    :STARTING ADDRESS OF WHERE
7105  :THE REGISTERS ARE SAVED
7106  18.      :NUMBER OF REGISTERS
7107  :SAVED = 18.
7108  JSR      PC, @CHECKT    :CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
7109  :AND THAT ALL STATUS BITS ARE = 0
7110  TYPE     .CPHALT        :CANNOT CONTINUE TESTING IF NOT
7111  HALT     :STOP TEST
7112  MOV      @RPHVEC, @RPVEC :SET RPO4 VECTOR ADDRESS
7113  :TO 'TIME1' IF P-CLOCK IS PRESENT
7114  :OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7115  :'TIME' WILL ONLY SAVE
7116  :CURRENT CYLINDER ADDRESS
7117  :AND LOOK AHEAD REGISTERS
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128  MOV      @WRIDAT, -(SP)    :GET READY TO MOVE COMMAND
7129  BIS      @GO!IE, (SP)     :GET READY TO SET GO AND
7130  :ENABLE INTERRUPT
7131  MOV      (SP)+, @RHCSI    :GO WITH
7132  :60 IN RHCSI FOR WRITE DATA
7133  :WITH INTERRUPT ENABLED
7134
7135
7136
7137  MOV      @TMP0, RO        :SET RO TO REG ADDRESS
7138  MOV      @BIT1!BIT10, @RO+ :ATTEMPT TO WRITE INTO A REGISTER
7139  :DURING WRITE DATA
7140  MOV      RO, @TMP0        :SAVE OFF NEW REG ADDRESS
7141
7142  : *NOW RMR MUST BE SET BUT THE COMPLETION OF
7143  : *WRITE DATA MUST BE WAITED ON
7144
7145
7146
7147  WAT      :WAIT FOR RDY BIT TO SET
7148  RHCSI    :WAIT FOR RHCSI REGISTER
7149  RDY      :WAIT FOR RDY BIT IN RHCSI REGISTER
7150  981.    :ALLOW 9810 MICRO SECONDS
7151  834.    :RDY MUST SET BETWEEN
7152  :1470 AND 18150 MICRO SECONDS
7153
7154  : *CHANGE SAVED REGISTERS TO EXPECTED VALUE
7155
7156  JSR      RO, @CHREG       :CHANGE BITS IN SAVED REGISTER
7157  RHCC     :CHANGE RHCC REGISTER
7158
7159  1        :1 BIT/BITS TO BE CHANGED
7160  1        :NEW VALUE OF BIT0 IS 1
7161  BIT0     :CHANGE BIT0 BIT
  
```

G16

```

7153 025254 004037 033674 JSR RO,0#CHREG ;CHANGE BITS IN SAVED REGISTER
7154 025260 002200 RHCS1 ;CHANGE RHCS1 REGISTER
7155 025262 000001 ;1 BIT/BITS TO BE CHANGED
7156 025264 000001 ;NEW VALUE OF SC IS 1
7158 025266 100000 SC ;CHANGE SC BIT
7160 025270 004037 033674 JSR RC,0#CHREG ;CHANGE BITS IN SAVED REGISTER
7161 025274 002222 RHDS1 ;CHANGE RHDS1 REGISTER
7162 025276 000002 ;2 BIT/BITS TO BE CHANGED
7163 025300 000001 ;NEW VALUE OF ATA IS 1
7164 025302 100000 ATA ;CHANGE ATA BIT
7165 025304 000001 ;NEW VALUE OF ERR IS 1
7166 025306 040000 ERR ;CHANGE ERR BIT
7168 025310 053737 004644 004536 BIS 0#ATTENT,0#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
7169 ;FOR WORKING DRIVE IN
7170 ;SAVED RHAS LOCATION
7171 025316 004037 033674 JSR RO,0#CHREG ;CHANGE BITS IN SAVED REGISTER
7172 025322 002202 RHER1 ;CHANGE RHER1 REGISTER
7173 025324 000001 ;1 BIT/BITS TO BE CHANGED
7174 025326 000001 ;NEW VALUE OF RMR IS 1
7175 025330 000004 RMR ;CHANGE RMR BIT
7176 025332 004037 032706 JSR RC,0#FILLRE ;MOV 1 INTO SAVED RHDST
7177 025336 002204 RHDST ;SAVED REGISTER TO CHANGE
7178 025340 000001 ;DATA
7179 025342 004037 032706 JSR RC,0#FILLRE ;MOV WRFROM+(256.*2) INTO SAVED RHBA
7180 025346 002174 RHBA ;SAVED REGISTER TO CHANGE
7181 025350 003370 WRFROM+(256.*2) ;DATA
7182 025352 004037 032706 JSR RC,0#FILLRE ;MOV 0 INTO SAVED RHWC
7183 025356 002172 RHWC ;SAVED REGISTER TO CHANGE
7184 025360 000000 ;DATA
7185 ;*COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
7186 ;*AFTER WRITE AND ATTEMPTED MODIFICATION OF REGISTER
7187 025362 004037 034002 JSR RO,0#COMREG ;COMPARE SAVED REGISTERS WITH
7188 ;PRESENT VALUE
7189 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7190 WC ;TEST DATA STARTING FROM 'RHWC'
7191 18. ;18. REGISTERS TO BE COMPARED
7192 25 ;RETURN TO 25 ON ERROR
7193 35 ;RETURN TO 35 ON NO ERROR
7200 025366 004512
7201 025370 002254
7202 025372 000022
7203 025374 025400
7204 025376 025420
    
```

19

Handwritten notes and signatures on the right side of the page.

H16

```

72206 025400          25:      MOV      RD, -(SP)          ;; PUSH RD ON STACK
72207 025400 010046      MOV      @#TMP0, RD        ;; GET REGISTER BEING MODIFIED + 2 POINTER
72208 025402 013700 004650  MOV      -(RD), @#SBDADR   ;; GET ADDRESS OF REGISTER BEING MODIFIED
72209 025406 014037 021122  ERROR    60              ;; ATTEMPTING TO MODIFY REGISTER
72210 025412 104060      MOV      (SP)+, RD        ;; POP STACK INTO RD
72211 025414 012600      RTS      PC              ;; DURING A WRITE COMMAND CAUSED
72212 025416 000207      ;; IMPROPER REGISTER GIVES WHAT SHOULD
72213      ;; GOOD DATA GIVES WHAT SHOULD
72214      ;; BE THERE
72215      ;; RECEIVED DATA GIVES WHAT WAS
72216      ;; THERE AFTER READ
72217
72218      ; *CLEAR ALL ERROR FLAGS
72219
72220 025420          35:
72221
72222 025420 004737 033004      JSR      PC, @#CLDISK     ;; SET R1-RHCS1, R2-RHCS2
72223      ;; R3-RHDS1, R4-RHER1
72224      ;; GIVE RH-11 INITIALIZE
72225      ;; SETUP UNIT NUMBER
72226
72227      ; *FILL WRITE FROM BUFFER WITH EXPECTED DATA
72228
72229 025434 004037 032654      JSR      RD, @#CLAREA    ;; CLEAR 256. WORDS, FROM WRFROM
72230 025430 002370      WRFROM      ;; STARTING FROM WRFROM
72231 025432 000400      256.        ;; 256. WORDS
72232 025434 002000      2000       ;; FILL WITH 2000
72233
72234
72235 025436 004037 032654      JSR      RD, @#CLAREA    ;; CLEAR 4 WORDS, FROM WRFROM+(256.*2)
72236 025442 003370      WRFROM+(256.*2) ;; STARTING FROM WRFROM+(256.*2)
72237 025444 000004      4           ;; 4 WORDS
72238 025446 002001      2001       ;; FILL WITH 2001
72239
72240
72241      ; *NOW THE READ DATA COMMAND WILL BE FILLED
72242
72243 025450 004037 034766      JSR      RD, @#RUN       ;; SETUP TO RUN FOR DATA COMMAND
72244 025454 000001      1           ;; CYLINDER 1
72245 025456 000      .BYTE 0        ;; SECTOR 0
72246 025457 000      .BYTE 0        ;; TRACK 0
72247 025460 177374      -260.      ;; WORD COUNT = 260.
72248 025462 003434      REINTO     ;; BUS ADDRESS
72249      ;; STARTING ADDRESS OF DATA
72250      ;; BUFFER = REINTO
72251 025464 000000      0           ;; DO NOT INHIBIT BUS ADDRESS INCREMENT
72252 025466 014000      ECI!FMT22 ;; 16 BITS PER WORD FORMAT
72253      ;; INHIBIT ECC CORRECTION
72254      ;; DO NOT INHIBIT HEADER COMPARE
72255 025470 002346      READAT    ;; GET READY TO DO A READAT
72256      ;; READ DATA WITH 70 IN RHCS1
72257
72258
72259      ; *NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

```

```

7260 025472 004037 033152 JSR RO,0#SAVER ;SAVE REGISTERS
7261 025476 002172 RHC ;RHC IS THE FIRST REGISTER SAVED
7262 025500 004512 SAVERE ;STARTING ADDRESS OF WHERE
7263 ;THE REGISTERS ARE SAVED
7264 025502 000022 18. ;NUMBER OF REGISTERS
7265 ;SAVED = 18.
7266
7267 025504 004737 033064 JSR PC,0#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
7268 ;AND THAT ALL STATUS BITS ARE = 0
7269 025510 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
7270 025514 000000 HALT ;STOP TEST
7271
7272 025516 013777 004506 15:42 MOV 0#RPO4VEC,0#RPVEC ;SET RPO4 VECTOR ADDRESS
7273 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7274 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7275 ;'TIME' WILL ONLY SAVE
7276 ;CURRENT CYLINDER ADDRESS
7277 ;AND LOOK AHEAD REGISTERS
7278
7279
7280 025524 013746 002346 MOV 0#READAT, -(SP) ;GET READY TO MOVE COMMAND
7281 025530 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
7282 ;ENABLE INTERRUPT
7283 025534 012677 154440 MOV (SP)+,0#RHCS1 ;GO WITH
7284 ;70 IN RHCS1 FOR READ DATA
7285 ;WITH INTERRUPT ENABLED
7286 025540 011100 MOV 0#R1, RO ;SAVE RHCS1 DURING ABOVE OPERATION
7287 025542 011305 MOV 0#R3, RE ;SAVE RHDS1 DURING ABOVE OPERATION
7288
7289
7290 025544 104412 WAT ;WAIT FOR RDY BIT TO SET
7291 025546 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
7292 025550 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7293 025552 001725 991. ;ALLOW 9910 MICRO SECONDS
7294 025554 001502 834. ;RDY MUST SET BETWEEN
7295 ;1470 AND 18150 MICRO SECONDS
7296
7297 ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7298 ;*RO AND R5 IMMEDIATELY AFTER GO IS ISSUED
7299
7300 025556 013746 002346 MOV 0#READAT, -(SP) ;SAVE COMMAND
7301 025562 052716 004101 BIS #IE!DVA!GO, (SP) ;INCLUDE IE!DVA!GO
7302 025566 011637 001124 MOV (SP), 0#SGDDAT ;SAVE FOR PRINTOUT
7303 025572 022600 CMP (SP)+, RO ;DURING ABOVE OPERATION ONLY IE!DVA!GO
7304 ;AND COMMAND SHOULD BE SET
7305 025574 001405 BEQ 70$ ;BRANCH IF GOOD
7306 025576 010037 001126 MOV RO, 0#SBDDAT ;BAD DATA
7307 025602 010137 004500 MOV R1, 0#REGADR ;FAILING REGISTER RHCS1
7308 025606 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
7309 ;COMMAND AND IE!DVA!GO SHOULD BE SET
7310 025610 012746 010500 70$: MOV #MOL!DPR!VV, -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7311 025614 011637 001124 MOV (SP), 0#SGDDAT ;SAVE FOR PRINTOUT
7312 025620 022605 CMP (SP)+, RE ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7313 ;SHOULD BE SET
    
```

J16

```

7314 025622 001405      BEQ      72$      :BRANCH IF GOOD
7315 025624 010537 001126    MOV      R5,2#$BDDAT :BAD DATA
7316 025630 010337 004500    MOV      R3,2#$REGADR :FADING REGISTER RHDS!
7317 025634 104063      ERROR    63        :DURING ABOVE OPERATION ONLY
7318                                     :MOL!DPR!VV SHOULD BE SET
7319 025636                                     72$:
7320
7321                                     :*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7322
7323 025636 004037 032706    JSR      RD,2#FILLRE  :MOV 0 INTO SAVED RHWC
7324 025642 002172      RHWC      :SAVED REGISTER TO CHANGE
7325 025644 000000      0        :DATA
7326
7327
7328
7329 025646 004037 032706    JSR      RD,2#FILLRE  :MOV REINTO+(260.*2) INTO SAVED RHBA
7330 025652 002174      RHBA      :SAVED REGISTER TO CHANGE
7331 025654 004444      REINTO+(260.*2) :DATA
7332
7333
7334 025656 004037 032706    JSR      RD,2#FILLRE  :MOV 2 INTO SAVED RHDST
7335 025662 002204      RHDST     :SAVED REGISTER TO CHANGE
7336 025664 000002      2        :DATA
7337
7338
7339                                     :*NOW COMPARE REGISTERS BEFORE READ DATA WITH
7340                                     :*AFTER COMMAND
7341
7342 025666 004037 034002    JSR      RD,2#COMREG  :COMPARE SAVED REGISTERS WITH
7343                                     :PRESENT VALUE
7344 025672 004512      SAVERE    :GOOD DATA SAVED IN 'SAVERE'
7345 025674 002254      WC        :TEST DATA STARTING FROM 'RHWC'
7346 025676 000022      18.      :18. REGISTERS TO BE COMPARED
7347 025700 025704      4$       :RETURN TO 4$ ON ERROR
7348 025702 025710      5$       :RETURN TO 5$ ON NO ERROR
7349
7350
7351 025704 104033 4$:      ERROR    33      :READ DATA CAUSED IMPROPER REGISTER
7352 025706 000207      RTS      PC      :CHANGE
7353                                     :GOOD DATA GIVES WHAT SHOULD BE THERE
7354                                     :RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
7355                                     :*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
7356                                     :*WAS GOOD
7357 025710 5$:
7358
7359 025710 004037 035032    JSR      RD,2#COMPAR  :COMPARE TWO BLOCKS OF MEMORY
7360 025714 002370      WRFROM   :GOOD DATA STARTS FROM WRFROM
7361 025716 003434      REINTO   :TEST DATA STARTS FROM REINTO
7362 025720 000400      256.    :256. WORDS TO BE COMPARED
7363 025722 025726      6$      :RETURN TO 6$ ON ERROR
7364 025724 025732      ST28    :RETURN TO ST28 ON NO ERROR
7365
7366
7367 025726 104034 6$:      ERROR    34      :READ DATA ERROR AFTER A WRITE DATA
  
```

K16

MNDEC-11-DZRJJ-A, RPO4 5/6 FUNCT. CONT. TST-PT 2 MACY11 27.655) 30-MAR-76 18:59 PAGE 156
DZRJJA.P11 T30 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2), 'RMR'

SEQ 0204

7368	025730	000207		RIS	PC		:WITH REGISTER MODIFICATION
7369							:WITHIN THE WRITE DATA
7370							:*IF ALL 7 REGISTERS NOT COMPLETE THEN REPEAT
7371	025732	005337	004656	ST28:	DEC	2#TMP5	:COUNT DOWN
7372	025736	001002			BNE	IS	:BRANCH IF 7 NOT DONE
7373	025740	000137	025750		JMP	TST31	:JUMP TO NEXT TEST
7374	025744	000137	024716	IS:	JMP	ST24	:JUMP TO BEGINING OF TEST
7375							

L16

```

7376
7377
7378
7379
7380
7381
7382
7383
7384
7385 025750 000004
7386 025752 012706 001000
7387 025756 012737 000031 004504
7388
7389 025764 004737 033004
7390
7391
7392
7393
7394
7395
7396 025770 004037 032654
7397 025774 002370
7398 025776 000400
7399 026000 000000
7400
7401
7402
7403 026002 004037 032654
7404 026006 003434
7405 026010 000400
7406 026012 177777
7407
7408
7409
7410 026014 004037 034766
7411 026020 000000
7412 026022 000
7413 026023 000
7414 026024 177400
7415 026026 003434
7416
7417
7418 026030 000000
7419 026032 014000
7420
7421
7422 026034 002346
7423
7424
7425
7426 026036 004037 033152
7427 026042 002172
7428 026044 004512
7429

```

```

*****
*TEST 31 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2)RMP
*****
* A READ DATA COMMAND IS GIVEN TO CYLINDER 0, SECTOR 0
* TRACK 0. IMMEDIATELY AFTER GO RHAS IS WRITTEN INTO
* WITH ALL ONES RMR BIT #2 IN RHER SHOULD NOT SET
*****
TST31: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #31,#TSTNM ;SAVE TEST NUMBER
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
JSR RD,#CLAREA ;CLEAR 256. WORDS. FROM WRFROM
WRFROM ;STARTING FROM WRFROM
256. ;256. WORDS
0 ;FILL WITH 0
;FILL READ INTM BUFFER WITH ALL ONES
JSR RD,#CLAREA ;CLEAR 256. WORDS. FROM REINTO
REINTO ;STARTING FROM REINTO
256. ;256. WORDS
-1 ;FILL WITH -1
;NOW THE READ DATA COMMAND WILL BE FILLED
JSR RC,#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
-256. ;WORD COUNT = 256.
REINTO ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = REINTO
;DO NOT INHIBIT BUS ADDRESS INCREMENT
;16 BITS PER WORD FORMAT
;INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
READAT ;GET READY TO DO A READAT
;READ DATA WITH 70 IN RHCS1
;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
JSR RO,#SAVER ;SAVE REGISTERS
RHWI ;RHWI IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED

```


M16

MNDEC-11-DZRJJ-A, RPO4/S/6 FUNCT. CONT. TST-PT 2 MACY11 27(655) 30-MAR-76 18:59 PAGE 158
 DZRJJA.P11 t31 REGISTER MODIFICATION REFUSED - RHER1 (BIT #2)RMR

SEQ 0206

```

7430 026046 000022          18.          ;NUMBER OF REGISTERS
7431                                ;SAVED = 18.
7432
7433 026050 004737 033064    JSR      PC,@#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
7434                                ;AND THAT ALL STATUS BITS ARE = 0
7435 026054 104400 056662    TYPE     .CPHALT     ;CANNOT CONTINUE TESTING IF NOT
7436 026060 000000          HALT      ;STOP TEST
7437
7438 026062 013777 004506 154076  MOV      @#RPO4VEC, @RPOVEC ;SET RPO4 VECTOR ADDRESS
7439                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
7440                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7441                                ;'TIME' WILL ONLY SAVE
7442                                ;CURRENT CYLINDER ADDRESS
7443                                ;AND LOOK AHEAD REGISTERS
7444
7445
7446
7447 026070 013746 002346    MOV      @#READAT, -(SP) ;GET READY TO MOVE COMMAND
7448 026074 052716 000101    BIS      #GO!IE, (SP)   ;GET READY TO SET GO AND
7449                                ;ENABLE INTERRUPT
7450 026100 012677 154074    MOV      (SP)+, @RHCS1 ;GO WITH
7451                                ;70 IN RHCS1 FOR READ DATA
7452                                ;WITH INTERRUPT ENABLED
7453 026104 011100          MOV      @R1, R0        ;SAVE RHCS1 DURING ABOVE OPERATION
7454 026106 011305          MOV      @R3, R5        ;SAVE RHDS1 DURING ABOVE OPERATION
7455
7456 026110 012777 177777 154100  MOV      #-1, @RHAS    ;WRITE INTO RHAS THIS SHOULD
7457                                ;NOT SET RMR
7458
7459                                ;*TIME IS NOT IMPORTANT
7460
7461 026116 104412          WAT      ;WAIT FOR RDY BIT TO SET
7462 026120 002200          RHCS1   ;WAIT FOR RHCS1 REGISTER
7463 026122 000200          RDY     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7464 026124 003326          1750.  ;ALLOW 17500 MICRO SECONDS
7465 026126 000175          125.   ;RDY MUST SET BETWEEN
7466                                ;16250 AND 18750 MICRO SECONDS
7467
7468                                ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7469                                ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
7470
7471 026130 013746 002346    MOV      @#READAT, -(SP) ;SAVE COMMAND
7472 026134 052716 004101    BIS      #IE!DVA!GO, (SP) ;INCLUDE IE!DVA!GO
7473 026140 011637 001124    MOV      (SP), @#SGDDAT ;SAVE FOR PRINTOUT
7474 026144 022600          CMP      (SP)+, R0      ;DURING ABOVE OPERATION ONLY IE!DVA!GO
7475                                ;AND COMMAND SHOULD BE SET
7476 026146 001405          BEQ     64$            ;BRANCH IF GOOD
7477 026150 010037 001126    MOV      R0, @#SBDDAT   ;BAD DATA
7478 026154 010137 004500    MOV      R1, @#REGADR   ;FAILING REGISTER RHCS1
7479 026160 104021          ERROR   21            ;DURING ABOVE OPERATION ONLY
7480                                ;COMMAND AND IE!DVA!GO SHOULD BE SET
7481 026162 012746 010500          64$: MOV      #MOL!DPR!VV, -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7482 026166 011637 001124    MOV      (SP), @#SGDDAT ;SAVE FOR PRINTOUT
7483 026172 022605          CMP      (SP)+, R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
  
```

75001
75002
75003
75004
75005
75006
75007
75008
75009
75010
75011
75012
75013
75014
75015
75016
75017
75018
75019
75020
75021
75022
75023
75024
75025
75026
75027
75028
75029
75030
75031
75032
75033
75034
75035
75036
75037

026200
026201
026202
026203
026204
026205
026206
026207
026208
026209
026210
026211
026212
026213
026214
026215
026216
026217
026218
026219
026220
026221
026222
026223
026224
026225
026226
026227
026228
026229
026230
026231
026232
026233
026234
026235
026236
026237
026238
026239
026240
026241
026242
026243
026244
026245
026246
026247
026248
026249
026250
026251
026252
026253
026254
026255
026256
026257
026258
026259
026260
026261
026262
026263
026264
026265
026266
026267
026268
026269
026270
026271
026272

JSR
RTS
ERROR
RTS
PC
JSR
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29
R30
R31
R32
R33
R34
R35
R36
R37
R38
R39
R40
R41
R42
R43
R44
R45
R46
R47
R48
R49
R50
R51
R52
R53
R54
R55
R56
R57
R58
R59
R60
R61
R62
R63
R64
R65
R66
R67
R68
R69
R70
R71
R72
R73
R74
R75
R76
R77
R78
R79
R80
R81
R82
R83
R84
R85
R86
R87
R88
R89
R90
R91
R92
R93
R94
R95
R96
R97
R98
R99
R100
R101
R102
R103
R104
R105
R106
R107
R108
R109
R110
R111
R112
R113
R114
R115
R116
R117
R118
R119
R120
R121
R122
R123
R124
R125
R126
R127
R128
R129
R130
R131
R132
R133
R134
R135
R136
R137
R138
R139
R140
R141
R142
R143
R144
R145
R146
R147
R148
R149
R150
R151
R152
R153
R154
R155
R156
R157
R158
R159
R160
R161
R162
R163
R164
R165
R166
R167
R168
R169
R170
R171
R172
R173
R174
R175
R176
R177
R178
R179
R180
R181
R182
R183
R184
R185
R186
R187
R188
R189
R190
R191
R192
R193
R194
R195
R196
R197
R198
R199
R200
R201
R202
R203
R204
R205
R206
R207
R208
R209
R210
R211
R212
R213
R214
R215
R216
R217
R218
R219
R220
R221
R222
R223
R224
R225
R226
R227
R228
R229
R230
R231
R232
R233
R234
R235
R236
R237
R238
R239
R240
R241
R242
R243
R244
R245
R246
R247
R248
R249
R250
R251
R252
R253
R254
R255
R256
R257
R258
R259
R260
R261
R262
R263
R264
R265
R266
R267
R268
R269
R270
R271
R272
R273
R274
R275
R276
R277
R278
R279
R280
R281
R282
R283
R284
R285
R286
R287
R288
R289
R290
R291
R292
R293
R294
R295
R296
R297
R298
R299
R300
R301
R302
R303
R304
R305
R306
R307
R308
R309
R310
R311
R312
R313
R314
R315
R316
R317
R318
R319
R320
R321
R322
R323
R324
R325
R326
R327
R328
R329
R330
R331
R332
R333
R334
R335
R336
R337
R338
R339
R340
R341
R342
R343
R344
R345
R346
R347
R348
R349
R350
R351
R352
R353
R354
R355
R356
R357
R358
R359
R360
R361
R362
R363
R364
R365
R366
R367
R368
R369
R370
R371
R372
R373
R374
R375
R376
R377
R378
R379
R380
R381
R382
R383
R384
R385
R386
R387
R388
R389
R390
R391
R392
R393
R394
R395
R396
R397
R398
R399
R400
R401
R402
R403
R404
R405
R406
R407
R408
R409
R410
R411
R412
R413
R414
R415
R416
R417
R418
R419
R420
R421
R422
R423
R424
R425
R426
R427
R428
R429
R430
R431
R432
R433
R434
R435
R436
R437
R438
R439
R440
R441
R442
R443
R444
R445
R446
R447
R448
R449
R450
R451
R452
R453
R454
R455
R456
R457
R458
R459
R460
R461
R462
R463
R464
R465
R466
R467
R468
R469
R470
R471
R472
R473
R474
R475
R476
R477
R478
R479
R480
R481
R482
R483
R484
R485
R486
R487
R488
R489
R490
R491
R492
R493
R494
R495
R496
R497
R498
R499
R500
R501
R502
R503
R504
R505
R506
R507
R508
R509
R510
R511
R512
R513
R514
R515
R516
R517
R518
R519
R520
R521
R522
R523
R524
R525
R526
R527
R528
R529
R530
R531
R532
R533
R534
R535
R536
R537
R538
R539
R540
R541
R542
R543
R544
R545
R546
R547
R548
R549
R550
R551
R552
R553
R554
R555
R556
R557
R558
R559
R560
R561
R562
R563
R564
R565
R566
R567
R568
R569
R570
R571
R572
R573
R574
R575
R576
R577
R578
R579
R580
R581
R582
R583
R584
R585
R586
R587
R588
R589
R590
R591
R592
R593
R594
R595
R596
R597
R598
R599
R600
R601
R602
R603
R604
R605
R606
R607
R608
R609
R610
R611
R612
R613
R614
R615
R616
R617
R618
R619
R620
R621
R622
R623
R624
R625
R626
R627
R628
R629
R630
R631
R632
R633
R634
R635
R636
R637
R638
R639
R640
R641
R642
R643
R644
R645
R646
R647
R648
R649
R650
R651
R652
R653
R654
R655
R656
R657
R658
R659
R660
R661
R662
R663
R664
R665
R666
R667
R668
R669
R670
R671
R672
R673
R674
R675
R676
R677
R678
R679
R680
R681
R682
R683
R684
R685
R686
R687
R688
R689
R690
R691
R692
R693
R694
R695
R696
R697
R698
R699
R700
R701
R702
R703
R704
R705
R706
R707
R708
R709
R710
R711
R712
R713
R714
R715
R716
R717
R718
R719
R720
R721
R722
R723
R724
R725
R726
R727
R728
R729
R730
R731
R732
R733
R734
R735
R736
R737
R738
R739
R740
R741
R742
R743
R744
R745
R746
R747
R748
R749
R750
R751
R752
R753
R754
R755
R756
R757
R758
R759
R760
R761
R762
R763
R764
R765
R766
R767
R768
R769
R770
R771
R772
R773
R774
R775
R776
R777
R778
R779
R780
R781
R782
R783
R784
R785
R786
R787
R788
R789
R790
R791
R792
R793
R794
R795
R796
R797
R798
R799
R800
R801
R802
R803
R804
R805
R806
R807
R808
R809
R810
R811
R812
R813
R814
R815
R816
R817
R818
R819
R820
R821
R822
R823
R824
R825
R826
R827
R828
R829
R830
R831
R832
R833
R834
R835
R836
R837
R838
R839
R840
R841
R842
R843
R844
R845
R846
R847
R848
R849
R850
R851
R852
R853
R854
R855
R856
R857
R858
R859
R860
R861
R862
R863
R864
R865
R866
R867
R868
R869
R870
R871
R872
R873
R874
R875
R876
R877
R878
R879
R880
R881
R882
R883
R884
R885
R886
R887
R888
R889
R890
R891
R892
R893
R894
R895
R896
R897
R898
R899
R900
R901
R902
R903
R904
R905
R906
R907
R908
R909
R910
R911
R912
R913
R914
R915
R916
R917
R918
R919
R920
R921
R922
R923
R924
R925
R926
R927
R928
R929
R930
R931
R932
R933
R934
R935
R936
R937
R938
R939
R940
R941
R942
R943
R944
R945
R946
R947
R948
R949
R950
R951
R952
R953
R954
R955
R956
R957
R958
R959
R960
R961
R962
R963
R964
R965
R966
R967
R968
R969
R970
R971
R972
R973
R974
R975
R976
R977
R978
R979
R980
R981
R982
R983
R984
R985
R986
R987
R988
R989
R990
R991
R992
R993
R994
R995
R996
R997
R998
R999
R1000

: SHOULD BE SET
: BRANCH IF GOOD
: SAC DATA
: FAILING REGISTER RHCS!
: DURING ABOVE OPERATION ONLY
: MCL! DPR! VV SHOULD BE SE

668:

: *NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

JSR RD,20FILLRE :MOV 0 INTO SAVED RHWC
RHWC :SAVED REGISTER TO CHANGE
0 :DATA

JSR RD,20FILLRE :MOV REINTO+(256.*2) INTO SAVED RHBA
RHBA :SAVED REGISTER TO CHANGE
REINTO+(256.*2) :DATA

JSR RD,20FILLRE :MOV 1 INTO SAVED RHDST
RHDST :SAVED REGISTER TO CHANGE
1 :DATA

JSR RD,20FILLRE :MOV 0 INTO SAVED RHCC
RHCC :SAVED REGISTER TO CHANGE
0 :DATA

: *NOW COMPARE REGISTERS BEFORE READ DATA WITH
: *AFTER COMMAND

JSR RD,20COMREG :COMPARE SAVED REGISTERS WITH
SAVERE :PRESENT VALUE
NO :GOOD DATA SAVED IN 'SAVERE'
18 :TEST DATA STARTING FROM 'RHWC'
19 :18. REGISTERS TO BE COMPARED
25 :RETURN TO 18 ON ERROR
26 :RETURN TO 25 ON NO ERROR

18:

ERROR 33 :READ DATA CAUSED IMPROPER REGISTER
RTS PC :CHANGE
GOOD DATA GIVES WHAT SHOULD BE THERE
RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND

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

25:

JSR RD,20COMPAR :COMPARE TWO BLOCKS OF MEMORY
WRFROM :GOOD DATA STARTS FROM WRFROM
REINTO :TEST DATA STARTS FROM REINTO
256. :256.. WORDS TO BE COMPARED

75999
75998
75997
75996
75995
75994
75993
75992
75991
75990
75989
75988
75987
75986
75985
75984
75983
75982
75981
75980
75979
75978
75977
75976
75975
75974
75973
75972
75971
75970
75969
75968
75967
75966
75965
75964
75963
75962
75961
75960
75959
75958
75957
75956
75955
75954
75953
75952
75951
75950
75949
75948
75947
75946
75945
75944
75943
75942
75941
75940
75939
75938
75937
75936
75935
75934
75933
75932
75931
75930
75929
75928
75927
75926
75925
75924
75923
75922
75921
75920
75919
75918
75917
75916
75915
75914
75913
75912
75911
75910
75909
75908
75907
75906
75905
75904
75903
75902
75901
75900
75899
75898
75897
75896
75895
75894
75893
75892
75891
75890
75889
75888
75887
75886
75885
75884
75883
75882
75881
75880
75879
75878
75877
75876
75875
75874
75873
75872
75871
75870
75869
75868
75867
75866
75865
75864
75863
75862
75861
75860
75859
75858
75857
75856
75855
75854
75853
75852
75851
75850
75849
75848
75847
75846
75845
75844
75843
75842
75841
75840
75839
75838
75837
75836
75835
75834
75833
75832
75831
75830
75829
75828
75827
75826
75825
75824
75823
75822
75821
75820
75819
75818
75817
75816
75815
75814
75813
75812
75811
75810
75809
75808
75807
75806
75805
75804
75803
75802
75801
75800
75799
75798
75797
75796
75795
75794
75793
75792
75791
75790
75789
75788
75787
75786
75785
75784
75783
75782
75781
75780
75779
75778
75777
75776
75775
75774
75773
75772
75771
75770
75769
75768
75767
75766
75765
75764
75763
75762
75761
75760
75759
75758
75757
75756
75755
75754
75753
75752
75751
75750
75749
75748
75747
75746
75745
75744
75743
75742
75741
75740
75739
75738
75737
75736
75735
75734
75733
75732
75731
75730
75729
75728
75727
75726
75725
75724
75723
75722
75721
75720
75719
75718
75717
75716
75715
75714
75713
75712
75711
75710
75709
75708
75707
75706
75705
75704
75703
75702
75701
75700
75699
75698
75697
75696
75695
75694
75693
75692
75691
75690
75689
75688
75687
75686
75685
75684
75683
75682
75681
75680
75679
75678
75677
75676
75675
75674
75673
75672
75671
75670
75669
75668
75667
75666
75665
75664
75663
75662
75661
75660
75659
75658
75657
75656
75655
75654
75653
75652
75651
75650
75649
75648
75647
75646
75645
75644
75643
75642
75641
75640
75639
75638
75637
75636
75635
75634
75633
75632
75631
75630
75629
75628
75627
75626
75625
75624
75623
75622
75621
75620
75619
75618
75617
75616
75615
75614
75613
75612
75611
75610
75609
75608
75607
75606
75605
75604
75603
75602
75601
75600
75599
75598
75597
75596
75595
75594
75593
75592
75591
75590
75589
75588
75587
75586
75585
75584
75583
75582
75581
75580
75579
75578
75577
75576
75575
75574
75573
75572
75571
75570
75569
75568
75567
75566
75565
75564
75563
75562
75561
75560
75559
75558
75557
75556
75555
75554
75553
75552
75551
75550
75549
75548
75547
75546
75545
75544
75543
75542
75541
75540
75539
75538
75537
75536
75535
75534
75533
75532
75531
75530
75529
75528
75527
75526
75525
75524
75523
75522
75521
75520
75519
75518
75517
75516
75515
75514
75513
75512
75511
75510
75509
75508
75507
75506
75505
75504
75503
75502
75501
75500
75499
75498
75497
75496
75495
75494
75493
75492
75491
75490
75489
75488
75487
75486
75485
75484
75483
75482
75481
75480
75479
75478
75477
75476
75475
75474
75473
75472
75471
75470
75469
75468
75467
75466
75465
75464
75463
75462
75461
75460
75459
75458
75457
75456
75455
75454
75453
75452
75451
75450
75449
75448
75447
75446
75445
75444
75443
75442
75441
75440
75439
75438
75437
75436
75435
75434
75433
75432
75431
75430
75429
75428
75427
75426
75425
75424
75423
75422
75421
75420
75419
75418
75417
75416
75415
75414
75413
75412
75411
75410
75409
75408
75407
75406
75405
75404
75403
75402
75401
75400
75399
75398
75397
75396
75395
75394
75393
75392
75391
75390
75389
75388
75387
75386
75385
75384
75383
75382
75381
75380
75379
75378
75377
75376
75375
75374
75373
75372
75371
75370
75369
75368
75367
75366
75365
75364
75363
75362
75361
75360
75359
75358
75357
75356
75355
75354
75353
75352
75351
75350
75349
75348
75347
75346
75345
75344
75343
75342
75341
75340
75339
75338
75337
75336
75335
75334
75333
75332
75331
75330
75329
75328
75327
75326
75325
75324
75323
75322
75321
75320
75319
75318
75317
75316
75315
75314
75313
75312
75311
75310
75309
75308
75307
75306
75305
75304
75303
75302
75301
75300
75299
75298
75297
75296
75295
75294
75293
75292
75291
75290
75289
75288
75287
75286
75285
75284
75283
75282
75281
75280
75279
75278
75277
75276
75275
75274
75273
75272
75271
75270
75269
75268
75267
75266
75265
75264
75263
75262
75261
75260
75259
75258
75257
75256
75255
75254
75253
75252
75251
75250
75249
75248
75247
75246
75245
75244
75243
75242
75241
75240
75239
75238
75237
75236
75235
75234
75233
75232
75231
75230
75229
75228
75227
75226
75225
75224
75223
75222
75221
75220
75219
75218
75217
75216
75215
75214
75213
75212
75211
75210
75209
75208
75207
75206
75205
75204
75203
75202
75201
75200
75199
75198
75197
75196
75195
75194
75193
75192
75191
75190
75189
75188
75187
75186
75185
75184
75183
75182
75181
75180
75179
75178
75177
75176
75175
75174
75173
75172
75171
75170
75169
75168
75167
75166
75165
75164
75163
75162
75161
75160
75159
75158
75157
75156
75155
75154
75153
75152
75151
75150
75149
75148
75147
75146
75145
75144
75143
75142
75141
75140
75139
75138
75137
75136
75135
75134
75133
75132
75131
75130
75129
75128
75127
75126
75125
75124
75123
75122
75121
75120
75119
75118
75117
75116
75115
75114
75113
75112
75111
75110
75109
75108
75107
75106
75105
75104
75103
75102
75101
75100
75099
75098
75097
75096
75095
75094
75093
75092
75091
75090
75089
75088
75087
75086
75085
75084
75083
75082
75081
75080
75079
75078
75077
75076
75075
75074
75073
75072
75071
75070
75069
75068
75067
75066
75065
75064
75063
75062
75061
75060
75059
75058
75057
75056
75055
75054
75053
75052
75051
75050
75049
75048
75047
75046
75045
75044
75043
75042
75041
75040
75039
75038
75037
75036
75035
75034
75033
75032
75031
75030
75029
75028
75027
75026
75025
75024
75023
75022
75021
75020
75019
75018
75017
75016
75015
75014
75013
75012
75011
75010
75009
75008
75007
75006
75005
75004
75003
75002
75001
75000

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

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

026314 000004
026316 012706 001000
026322 012737 000032 004504

TST32: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #32,2#1STNM ;SAVE TEST NUMBER
JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

026330 004737 033004
026334 005737 004640
026340 001402
026342 000137 027142
026346

TST 2#RH70 ;RH70 CONTROLLER ?
BEG 30\$;SKIP NEXT IF NOT = 1
JMP TST33 ;IF SO SKIP THIS TEST -----)

30\$:

;*GENERATE ILLEGAL FUNCTION

026346 005037 001200
026352 012700 002322 1\$:
026356 012705 000021
026362 023720 001200 2\$:
026366 001004
026370 062737 000002 001200
026376 000765
026400 005305 3\$:
026402 001367
026404 032737 000100 001200
026412 001001
026414 000402
026416 000137 027142 20\$:
026422 013737 001200 002364 19\$:
026430 062737 000002 001200

CLR 2#STMP1 ;GET READY TO MAKE ILLEGAL FUNCTION
MOV #FUTABL,RC ;GET POINTER TO BEGINNING OF COMMANDS
MOV #17,R5 ;COUNTER (17 GOOD FUNCTIONS)
CMP 2#STMP1,(RC)+ ;IS THIS A LEGAL FUNCTION
BNE 3\$;BRANCH IF NOT LEGAL
ADD #2,2#STMP1 ;MAKE ANOTHER FUNCTION
BR 1\$;GET READY TO TEST NEW FUNCTION
DEC R5 ;NOT LEGAL SO DECREMENT COUNTER
BNE 2\$;BRANCH IF 17 NOT DONE
BIT #100,2#STMP1 ;ALL BITS UP TO BIT #5 COMPARED ?
BNE 20\$;BRANCH OUT IF DONE
BR 19\$;BRANCH TO CONTINUE
JMP 2#7\$;DONE
MOV 2#STMP1,2#ILLEGL ;AN ILLEGAL FUNCTION IS FOUND
ADD #2,2#STMP1 ;GET READY FOR NEW FUNCTION NEXT TIME

;*ILLEGAL FUNCTION HAS BEEN FOUND

;*IT IS IN 'ILLEGL'
MOV #45,2#SLPERR ;ERROR RETURN POINT

;*SAVE REGISTERS FOR COMPARISON AFTER GO

026444
026444 004737 033004

4\$:
JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

7600	026450	005077	153516		CLR	2RHWC	: CLEAR WORD COUNT
7601	026454	005077	153514		CLR	2RHBA	: CLEAR BUS ADDRESS
7602	026460	023727	002364	000050	CMP	2#ILLEGL, #50	: 50 AND HIGHER FUNCTIONS ARE DATA
7603							: FUNCTIONS WHICH WILL SET MXF AND TRE
7604	026466	103014			BHIS	13\$: BRANCH IF ILLEGL IS HIGHER THAN 50
7605	026470	012737	100000	027044	MOV	#SC, 2#11\$+12	: EXPECTED VALUE OF RHCS1 SHOULD HAVE
7606							: ONLY SC ADDED
7607	026476	005037	027066		CLR	2#12\$+12	: EXPECTED VALUE OF RHCS2 SHOULD HAVE
7608							: NOTHING ADDED
7609	026502	005037	027072		CLR	2#12\$+16	: NO BITS TO BE CLEARED IN RHCS2
7610	026506	005037	027102		CLR	2#15\$+6	: RHBA SHOULD BE 0
7611	026512	005037	027112		CLR	2#16\$+6	: CLEAR SAVED RHWC
7612	026516	000500			BR	14\$: BRANCH
7613	026520	022737	000064	002364	13\$: CMP	#64, 2#ILLEGL	: IS FUNCTION 64
7614	026526	001020			BNE	17\$: BRANCH IF NOT
7615	026530	012737	140000	027044	MOV	#SC!TRE, 2#11\$+12	: SAVED RHCS1 SHOULD HAVE SC AND TRE
7616	026536	012737	000204	027102	MOV	#204, 2#15\$+6	: RHBA SHOULD HAVE 204
7617	026544	012737	000102	027112	MOV	#102, 2#16\$+6	: RHWC SHOULD HAVE 102
7618	026552	012737	001200	027066	MOV	#MXF!OR, 2#12\$+12	: RHCS2 SHOULD HAVE MXF AND OR
7619	026560	012737	000100	027072	MOV	#IR, 2#12\$+16	: RHCS2 SHOULD HAVE IR CLEARED
7620	026566	000454			BR	14\$: BRANCH
7621	026570	022737	000066	002364	17\$: CMP	#66, 2#ILLEGL	: IS FUNCTION 66
7622	026576	001030			BNE	18\$: BRANCH IF NOT
7623	026600	012777	177672	153364	MOV	#-70, 2RHWC	: MOVE 70 INTO RHWC
7624	026606	012777	002370	153360	MOV	#WRFROM, 2RHBA	: FILL RHBA WITH WRFROM
7625	026614	012737	140000	027044	MOV	#SC!TRE, 2#11\$+12	: SAVED RHCS1
7626	026622	012737	002164	027102	MOV	#WRFROM-(66.*2), 15\$+6: RHBA	
7627	026630	012737	177774	027112	MOV	#-4, 16\$+6	: SAVED RHWC
7628	026636	012737	001200	027066	MOV	#MXF!OR, 2#12\$+12	: SAVED RHCS2
7629	026644	005037	027072		CLR	2#12\$+16	: RHCS2
7630	026650	012737	000100	027072	MOV	#IR, 2#12\$+16	: RHCS2 SHOULD HAVE IR CLEARED
7631	026656	000420			BR	14\$: BRANCH
7632	026660	005077	153306		18\$: CLR	2RHWC	: CLEAR RHWC
7633	026664	005077	153304		CLR	2RHBA	: CLEAR RHBA
7634	026670	012737	140000	027044	MOV	#SC!TRE, 2#11\$+12	: RHCS1 SHOULD HAVE SC AND TRE
7635	026676	005037	027102		CLR	2#15\$+6	: RHBA
7636	026702	005037	027112		CLR	2#16\$+6	: RHWC
7637	026706	012737	001000	027066	MOV	#MXF, 2#12\$+12	: RHCS2
7638	026714	005037	027072		CLR	2#12\$+16	: RHCS2
7639	026720				14\$: JSR	RO, 2#SAVER	: SAVE REGISTERS
7640	026720	004037	033152				: RHWC IS THE FIRST REGISTER SAVED
7641	026724	002172			RHWC		: STARTING ADDRESS OF WHERE
7642	026726	004512			SAVERE		: THE REGISTERS ARE SAVED
7643							: NUMBER OF REGISTERS
7644	026730	000022			18.		: SAVED = 18.
7645							
7646							
7647	026732	004737	033064		JSR	PC, 2#CHECKT	: CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
7648							: AND THAT ALL STATUS BITS ARE = 0
7649	026736	104400	056662		TYPE	,CPHALT	: CANNOT CONTINUE TESTING IF NOT
7650	026742	000000			HALT		: STOP TEST
7651	026744	012746	002364		MOV	2#ILLEGL, -(SP)	: GET ILLEGAL FUNCTION
7652	026750	052716	000101		BIS	#GO!IE, (SP)	: INCLUDE IE AND GO
7653	026754	012611			MOV	(SP)+, JRI	: GO TO RHCS1 WITH ILLEGAL FUNCTION

```

7654
7655 026756 104412 WAT ;WAIT FOR RDY BIT TO SET
7656 026760 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
7657 026762 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7658 026764 001614 908. ;ALLOW 9080 MICRO SECONDS
7659 026766 001613 907. ;RDY MUST SET BETWEEN
7660 ;10 AND 18150 MICRO SECONDS
7661
7662 ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
7663
7664 026770 004037 033674 JSR RC,2*CHREG ;CHANGE BITS IN SAVED REGISTER
7665 026774 002202 RHCS1 ;CHANGE RHER1 REGISTER
7666
7667 026776 000001 1 ;1 BIT/BITS TO BE CHANGED
7668 027000 000001 ;NEW VALUE OF ILF IS 1
7669 027002 000001 ILF ;CHANGE ILF BIT
7670
7671 027004 004037 033674 JSR RC,2*CHREG ;CHANGE BITS IN SAVED REGISTER
7672 027010 002222 RHCS1 ;CHANGE RHCS1 REGISTER
7673
7674 027012 000002 2 ;2 BIT/BITS TO BE CHANGED
7675 027014 000001 ;NEW VALUE OF ATA IS 1
7676 027016 100000 ATA ;CHANGE ATA BIT
7677 027020 000001 ;NEW VALUE OF ERR IS 1
7678 027022 040000 ERR ;CHANGE ERR BIT
7679
7680 027024 053737 004544 004536 BIS 2*ATTENT,2*SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
7681 ;FOR WORKING DRIVE IN
7682 ;SAVED RHAS LOCATION
7683
7684 ;*RHCS1 WILL HAVE SC AND TRE ADDED IF FUNCTION IS GREATER THAN SC
7685 027032 125:
7686
7687 027032 004037 033674 JSR RC,2*CHREG ;CHANGE BITS IN SAVED REGISTER
7688 027036 002200 RHCS1 ;CHANGE RHCS1 REGISTER
7689
7690 027040 000001 1 ;1 BIT/BITS TO BE CHANGED
7691 027042 000001 ;NEW VALUE OF SC IS 1
7692 027044 100000 SC ;CHANGE SC BIT
7693 027046 053737 002364 004520 BIS 2*ILLEGL,2*SAVERE+6 ;INCLUDE ILLEGAL FUNCTION
7694 ;IN RHCS1
7695 ;*RHCS2 WILL HAVE NOTHING ADDED IF FUNCTION IS LESS THAN SC
7696 027054 125:
7697
7698 027054 004037 033674 JSR RC,2*CHREG ;CHANGE BITS IN SAVED REGISTER
7699 027060 002176 RHCS2 ;CHANGE RHCS2 REGISTER
7700
7701 027062 000002 2 ;2 BIT/BITS TO BE CHANGED
7702 027064 000001 ;NEW VALUE OF MXF IS 1
7703 027066 001000 MXF ;CHANGE MXF BIT
7704 027070 000000 0 ;NEW VALUE OF IR IS 0
7705 027072 000100 IR ;CHANGE IR BIT
7706
7707 027074 153:

```



```

7708 027074 004037 032706 JSR RO.3#FILLRE ;MOV 0 INTO SAVED RHBA
7709 027100 002174 RHBA ;SAVED REGISTER TO CHANGE
7710 027102 000000 0 ;DATA
7711 027104 165:
7712 027104 004037 032706 JSR RO.3#FILLRE ;MOV 0 INTO SAVED RHWC
7713 027110 002172 RHWC ;SAVED REGISTER TO CHANGE
7714 027112 000000 C ;DATA
7715
7716 ;*NOW COMPARE REGISTERS AFTER GIVING AN ILLEGAL COMMAND
7717
7718 027114 004037 034002 JSR RO.5#COMREG ;COMPARE SAVED REGISTERS WITH
7719 ;PRESENT VALUE
7720 027120 004512 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7721 027122 002254 WC ;TEST DATA STARTING FROM 'RHWC'
7722 027124 000022 LA ;18. REGISTERS TO BE COMPARED
7723 027126 027132 JSR ;RETURN TO 55 ON ERROR
7724 027130 027136 65 ;RETURN TO 65 ON NO ERROR
7725
7726
7727
7728
7729
7730 027132 104051 55: ERROR 31 ;GIVING ILLEGAL FUNCTION CAUSED
7731 027134 000207 RTS PC ;IMPROPER REGISTER CHANGE
7732 ;GOOD DATA GIVES WHAT
7733 ;SHOULD BE THERE
7734 ;RECEIVED DATA GIVES REGISTER
7735 ;CONTENTS AFTER ILLEGAL
7736 ;FUNCTION WA GIVEN
7737 027136 000137 026352 65: JMP 0#15 ;BRANCH FOR NEXT FUNCTION
7738 027142 75:
7739
7740 027142 105:
7741
7742
7743
7744
7745

```

1

ILLEGAL FUNCTION RHER1 - (BIT #0,ILF)

7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797

027142 000004
027144 012706 001000
027150 012737 000033 004504
027156 004737 033004
027162 012777 177374 153002
027170 012700 002370
027174 010077 152774
027200 012710 010000
027204 012720 000001
027210 005020
027212 005020
027214 012705 000400
027220 012730 177777
027224 005305
027226 001374
027230 012777 000001 152746
027236 004737 033064
027242 104400 056662
027246 000000
027250 013711 002344
027254 005037 004632
027260 012777 010000 152722
027266 005077 152720
027272 004037 033152
027276 002172
027300 004512
027302 000023

```
*****  
*TEST 33 OPERATION INCOMPLETE - RHER1(BIT #12)CPI  
* A WRITE HEADER AND DATA COMMAND IS GIVEN  
* CYLINDER 0 SECTOR 1 TRACK 0 KEYS 0 DATA 177777  
* WORDCOUNT 260  
*  
* AFTER GO IS GIVEN THEN THREE INDEX PULSES ARE  
* GIVEN. THIS SHOULD BRING CPI HIGH  
*****  
TST33: SCOPE  
MOV #STACK, SP ;RESET STACK  
MOV #33, #TSTNM ;SAVE TEST NUMBER  
  
JSR PC, #OLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
  
;*THESE ARE REGULAR SETUPS  
MOV #-260, #RHWC ;256 DATA WORDS 4 HEADER WORDS  
MOV #WRFROM, R0 ;THESE TWO INSTRUCTIONS GETS  
MOV R0, #RHBA ;ADDR. OF WRFROM BUFFER INTO R0 AND  
;BUS ADDRESS REGISTER  
MOV #FMT22, (R0); ;FORMAT=16 BIT WORDS  
;CYLINDER=0  
MOV #1, (R0)+ ;TRACK=0, SECTOR=1, KEYS=0  
CLR (R0)+ ;KEY1=0  
CLR (R0)+ ;KEY2=0  
MOV #256, R5 ;COUNTER  
  
15: ;*SETUP DATA, WRITE HEADER & DATA, AND FORMAT OF THE WRITE  
MOV #-1, (R0)+ ;MOVE ALL ONES FOR DATA  
DEC R5  
BNE 15 ;BRANCH IF DATA NOT COMPLETE  
MOV #1, #RHDS1 ;TRACK=0 SECTOR=1  
JSR PC, #CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1  
;AND THAT ALL STATUS BITS ARE = 0  
;CANNOT CONTINUE TESTING IF NOT  
;STOP TEST  
TYPE ,CPHALT  
HALT  
MOV #WRIFOR, #R1 ;GET READY FOR WRITE HEADER AND  
;DATA WITH 62 IN RHCS1  
CLR #ERFLGS ;CLEAR ERROR FLAG  
MOV #FMT22, #RHOF ;FORMAT BIT=1 (16 BIT WORDS)  
CLR #RHCA ;CYLINDER =0  
  
;*SAVE REGISTERS FOR COMPARISON AFTER READ  
JSR R0, #SAVER ;SAVE REGISTERS  
RHC ;RHC IS THE FIRST REGISTER SAVED  
SAVER ;STARTING ADDRESS OF WHERE  
;THE REGISTERS ARE SAVED  
19. ;NUMBER OF REGISTERS  
;SAVED = 19.
```


I01

```

7799
7800
7801
7802
7803
7804 027304 013700 002220      MOV      2#RHMR,RO      ;NOW RO HAS MAINTENANCE REG. ADDR.
7805 027310 012710 000001      MOV      #DMD,2RO      ;SET DIAGNOSTIC MODE
7806 027314 052710 000004      BIS      #MINX,2RO      ;SET INDEX
7807 027320 042710 000004      BIC      #MINX,2RO      ;CLEAR INDEX THIS GIVES
7808                                     ;ONE INDEX PULSE
7809
7810 027324 052777 000001 152646      BIS      #GO,2RHCS1     ;ISSUE THE 'GO' BIT TO THE RH11
7811 027332 012737 000113 004642      RUNWAT: MOV      #75.,2#RUNCTR ;LOAD 'RUN' LINE DELAY COUNTER
7812                                     ;= APPROX 450 US ON 11/50 CPU WITH CCRE
7813                                     ;AND PROVIDES FOR TIME TO FILL THE SILO
7814 027340 005337 004642      1$:      DEC      2#RUNCTR     ;COUNT DOWN ONCE
7815 027344 001375                                     ;CONTINUE UNTIL = 0
7816
7817                                     ;*ISSUE THE FIRST DIAGNOSTIC INDEX PULSE
7818 027346 052710 000004      BIS      #MINX,2RO      ;SET INDEX PULSE
7819 027352 042710 000004      BIC      #MINX,2RO      ;RESET INDEX
7820
7821                                     ;*SECOND INDEX PULSE
7822 027356 052710 000004      BIS      #MINX,2RO      ;SET INDEX
7823 027362 042710 000004      BIC      #MINX,2RO      ;CLEAR INDEX
7824
7825                                     ;*THIRD INDEX PULSE
7826 027366 052710 000004      BIS      #MINX,2RO      ;SET INDEX
7827 027372 042710 000004      BIC      #MINX,2RO      ;CLEAR INDEX
7828
7829
7830                                     ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
7831
7832 027376 004037 033674      JSR      RO,2#CHREG     ;CHANGE BITS IN SAVED REGISTER
7833 027402 002200      RHCS1    ;CHANGE RHCS1 REGISTER
7834
7835 2                                     ;2 BIT/BITS TO BE CHANGED
7836 1                                     ;NEW VALUE OF SC IS 1
7837 SC                                     ;CHANGE SC BIT
7838 1                                     ;NEW VALUE OF TRE IS 1
7839 1                                     ;CHANGE TRE BIT
7840 TRE
7841 027416 004037 033674      JSR      RO,2#CHREG     ;CHANGE BITS IN SAVED REGISTER
7842 027422 002222      RHDS1    ;CHANGE RHDS1 REGISTER
7843
7844 2                                     ;2 BIT/BITS TO BE CHANGED
7845 1                                     ;NEW VALUE OF ATA IS 1
7846 ATA                                     ;CHANGE ATA BIT
7847 1                                     ;NEW VALUE OF ERR IS 1
7848 1                                     ;CHANGE ERR BIT
7849 ERR
7850 027436 004037 032706      JSR      RO,2#FILLRE    ;MOV 2 INTO SAVED RHDST
7851 027442 002204      RHDST    ;SAVED REGISTER TO CHANGE
  
```

```

7852 027444 000002          2          ;DATA
7853
7854
7855 027446 004037 033674 JSR      RD,2#CHREG      ;CHANGE BITS IN SAVED REGISTER
7856 027452 002202          RHER1          ;CHANGE RHER1 REGISTER
7857
7858 027454 000001          1          ;1 BIT/BITS TO BE CHANGED
7859 027456 000001          1          ;NEW VALUE OF OPI IS 1
7860 027460 020000          CPI          ;CHANGE OPI BIT
7861
7862 027462 053737 004644 004536 BIS      2#ATTENT,2#SAVERE+24 ;SET APPROPRIATE 'ATA' BITS
7863                                     ;FOR WORKING DRIVE IN
7864                                     ;SAVED RHAS LOCATION
7865
7866 027470 004037 033674 JSR      RD,2#CHREG      ;CHANGE BITS IN SAVED REGISTER
7867 027474 002220          RHMR          ;CHANGE RHMR REGISTER
7868
7869 027476 000001          1          ;1 BIT/BITS TO BE CHANGED
7870 027500 000001          1          ;NEW VALUE OF DMD IS 1
7871 027502 000001          DMD          ;CHANGE DMD BIT
7872
7873
7874                                     ;*RHWC,RHBA AND OR AND IR BITS OF RHCS2 WILL NOT BE CHECKED
7875 027504 017737 152462 004512 MOV      2#RHWC,2#SAVERE ;SAVED RHWC
7876 027512 017737 152456 004514 MOV      2#RHBA,2#SAVERE+2 ;SAVED RHBA
7877 027520 017746 152452          MOV      2#RHCS2-(SP) ;GET RHCS2
7878 027524 042716 177477          BIC      #1<IR!OR>(SP) ;GET 'IR' & 'OR' STATES
7879 027530 042737 000300 004516 BIC      #IR!OR,2#SAVERE+4 ;CLEAR 'IR' & 'OR' BITS
7880 027536 052637 004516 BIS      (SP)+,2#SAVERE+4 ;SET 'OR' & 'IR' AS REQUIRED
7881
7882
7883                                     ;*COMPARE REGISTERS BEFORE WRITE WITH RESULTS AFTER WRITE
7884
7885 027542 004037 034002 JSR      RD,2#COMREG      ;COMPARE SAVED REGISTERS WITH
7886                                     ;PRESENT VALUE
7887 027546 004512          SAVERE         ;GOOD DATA SAVED IN 'SAVERE'
7888 027550 002254          WC          ;TEST DATA STARTING FROM 'RHWC'
7889 027552 000021          17.         ;17. REGISTERS TO BE COMPARED
7890 027554 027560          2$          ;RETURN TO 2$ ON ERROR
7891 027556 027564          3$          ;RETURN TO 3$ ON NO ERROR
7892
7893
7894 027560 104071          2$:      ERROR 71      ;FORCING OPI CAUSED
7895 027562 000207          RTS      PC      ;IMPROPER REGISTER CHANGE
7896                                     ;GOOD DATA GIVES WHAT SHOULD BE THERE
7897                                     ;RECEIVED DATA GIVES WHAT WAS THERE
7898                                     ;AFTER 3 INDEX PULSES WERE ISSUED
7899
7900
7901 027564 004737 033004          3$:      JSR      PC,2#CLDISK ;CLEAR THE 'GO' BIT
    
```

```

7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916 027570 000004
7917 027572 012706 001000
7918 027576 012737 000034 004504
7919
7920 027604 004737 033004
7921
7922
7923
7924 027610 012737 000025 027644
7925 027616 012737 000025 027660
7926 027624 012737 000056 004652
7927
7928
7929 027632
7930
7931 027632 004037 032630
7932 027636 002370
7933 027640 000004
7934 027642 010000
7935 027644 000025
7936 027646 000000
7937 027650 000000
7938
7939
7940 027652
7941
7942 027652 004037 034766
7943 027656 000000
7944 027660 025
7945 027661 000
7946 027662 177774
7947
7948 027664 002370
7949
7950
7951 027666 000000
7952 027670 010000
7953
7954
7955 027672 002344

```

```

*****
*TEST 34 CONSECUTIVE SECTOR FORMATTING
*
* 46 CONSECUTIVE SECTORS WILL BE FORMATTED
* STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
* FORMATTING WILL BE DONE BY A WRITE HEADER AND
* DATA COMMAND FOR 4 WORDS, ONE SECTOR
* AT A TIME.
*
* AFTER EACH SECTOR IS WRITTEN, 'SC' WILL BE CHECKED
* TO INSURE THAT THERE WERE NO ERRORS
*****
TST34: SCOPE
MOV #STACK.SP ;RESET STACK
MOV #34, @TSTNM ;SAVE TEST NUMBER
JSR PC, @CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
MOV #21., @#1$+12 ;SET UP TO START FROM
MOV #21., @#2$+6 ;SECTOR 21.
MOV #46., @#TMP1 ;46 SECTORS TO COVER 3 TRACKS
; *FILL WRITE FROM BUFFER WITH THE HEADER
1$: JSR RO, @#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
21. ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
; *NOW THE WRITE HEADER AND DATA COMMAND WILL BE SETUP
2$: JSR RO, @#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 21. ;SECTOR 21.
.BYTE 0 ;TRACK 0
-0-4 ;WORD COUNT (DATA) = 0 +
;4 HEADER WORDS
WRFROM ;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM
0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
FMT22 ;16 BITS PER WORD FORMAT
;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
WRFOR ;GET READY TO DO A WRIFOR

```

;WRITE HEADER AND DATA WITH 62 IN RHCS1

```

7956
7957
7958
7959 027674 004737 033064 JSR PC, @CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
7960 ;AND THAT ALL STATUS BITS ARE = 0
7961 027700 104400 055662 TYPE .CPHALT ;CANNOT CONTINUE TESTING IF NOT
7962 027704 000000 HALT ;STOP TEST
7963
7964 027706 013777 004506 152252 MOV @*RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
7965 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7966 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7967 ;'TIME' WILL ONLY SAVE
7968 ;CURRENT CYLINDER ADDRESS
7969 ;AND LOOK AHEAD REGISTERS
7970
7971
7972 027714 013746 002344 MOV @*WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
7973 027720 052716 000101 BIS @GO!IE, (SP) ;GET READY TO SET GO AND
7974 ;ENABLE INTERRUPT
7975 027724 012677 152250 MOV (SP)+, @RHCS1 ;GO WITH
7976 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7977 ;WITH INTERRUPT ENABLED
7978 027730 011100 MOV @R1, RC ;SAVE RHCS1 DURING ABOVE OPERATION
7979 027732 011305 MOV @R3, R5 ;SAVE RHDS1 DURING ABOVE OPERATION
7980
7981 ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
7982
7983 027734 104412 WAT ;WAIT FOR RDY BIT TO SET
7984 027736 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
7985 027740 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7986 027742 003237 1695. ;ALLOW 16950 MICRO SECONDS
7987 027744 001515 845. ;RDY MUST SET BETWEEN
7988 ;8500 AND 25400 MICRO SECONDS
7989
7990 ;*NOW ONE MORE SECTOR HAS BEEN WRITTEN
7991 ;*'SC' WILL BE CHECKED TO MAKE SURE
7992 ;*NO ERRORS OCCURED
7993
7994 027746 017737 152226 002262 MOV @RHCS1, @*CS1 ;GET RHCS1
7995 027754 032737 100000 002262 BIT @*SC, @*CS1 ;IS 'SC' SET ?
7996 027762 001403 BEQ 35 ;BRANCH IF "SPECIAL CONDITION" NOT SET
7997 027764 004737 034726 JSR PC, @*PUTREG ;READ & SAVE ALL RH11 & RPO4 REGISTERS
7998 027770 104072 ERROR 72 ;THERE WAS AN UNDEFINED ERROR AFTER
7999 ;A WRITE HEADER AND DATA
8000
8001 ;*A SECTOR HAS BEEN FORMATTED NOW
8002 ;*THE HARDWARE WILL BE CLEARED AND
8003 ;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
8004
8005 027772 35:
8006
8007 027772 004737 033004 JSR PC, @*CLDISK ;SET R1-RHCS1, R2-RHCS2
8008 ;R3-RHDS1, R4-RHER1
8009 ;GIVE R4-11 INITIALIZE
    
```

```

8010
8011 027776 013705 027644      MOV      2#15+12,RS      ;SETUP UNIT NUMBER
8012 030002 005205              INC      RS              ;GET SECTOR TRACK WORD
8013 030004 122705 000026      CMPB    #22,RS          ;+ 1
8014 030010 001405              BEQ     4$              ;IS IT 22 SECTORS (WHOLE TRACK DONE) ?
8015 030012 010537 027644      MOV     RS,2#15+12     ;YES...DO NEXT TRACK
8016 030016 010537 027660      MOV     RS,2#25+6     ;NO...RESTORE SECTOR TRACK FOR DATA
8017 030022 000410              BR      5$              ;RESTORE SECTOR TRACK FOR "RUN" ROUTINE
8018
8019 030024 105037 027644      4$:     CLRB   2#15+12   ;SET SECTOR = 0 FOR DATA WRITTEN
8020 030030 105237 027645      INCB   2#15+13         ;INCR TRACK FOR DATA WRITTEN
8021 030034 105037 027660      CLRB   2#25+6         ;SET SECTOR = 0 FOR "RUN" ROUTINE
8022 030040 105237 027661      INCB   2#25+7         ;INCR TRACK FOR THE "RUN" ROUTINE
8023
8024 030044 005337 004652      5$:     DEC     2#TMP1    ;ARE 46 SECTORS DONE ?
8025 030050 001270              BNE    1$              ;CONTINUE FORMATTING IF NOT
8026
8027 030052              6$:

```

;GO ON TO NEXT TEST IF SO

8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081

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

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

* FIRST 46 CONSECUTIVE SECTORS WILL BE FORMATTED
* STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
* FORMATTING WILL BE DONE BY A WRITE HEADER AND
* DATA COMMAND FOR 4 WORDS, ONE SECTOR
* AT A TIME

* THEN A READ HEADER AND DATA WILL BE DONE
* FOR CYLINDER 0 TRACK 0 SECTOR 21 FOR
* 11960 WORDS (260.X22X2+260+260) WITH BUS
* ADDRESS INHIBIT SET.

* AT THE END ALL REGISTERS WILL BE CHECKED.

030052 000004
030054 012706 001000
030060 012737 000035 004504
030066 004737 033004
030072 012737 000025 030126
030100 012737 000025 030142
030106 012737 000056 004652
030114
030114 004037 032630
030120 002370
030122 000004
030124 010000
030126 000025
030130 000000
030132 000000
030134
030134 004037 034756
030140 000000
030142 025
030143 000
030144 177774

TST35: SCOPE
MOV #STACK.SP ;RESET STACK
MOV #35,0#TSTNM ;SAVE TEST NUMBER
JSR PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
MOV #21,0#15+12 ;SET UP TO START FROM
MOV #21,0#25+5 ;SECTOR 21.
MOV #46,0#TMP1 ;46 SECTORS TO COVER 3 TRACKS

15: ;*FILL WRITE FROM BUFFER WITH HEADER

JSR RO,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
21. ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD

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

JSR RO,0#RUN ;SETUP TO RUN FOR DATA COMMAND
0 ;CYLINDER 0
.BYTE 21. ;SECTOR 21.
.BYTE 0 ;TRACK 0
-0-4 ;WORD COUNT (DATA) = 0 +
;4 HEADER WORDS

5 6 FUNCT CONT TEST-PT 3
OPERATION INCOMPLETE - RHER: BIT 013 OF:

```

030146 002370 WAFROM :BUS ADDRESS
:STARTING ADDRESS OF DATA
:BUFFER = WAFROM
030153 000000 0 :DO NOT INHIBIT BUS ADDRESS INCREMENT
030153 010000 0 FMT22 :6 BITS PER WORD FORMAT
:DO NOT INHIBIT ECC CORRECTION
:DO NOT INHIBIT HEADER COMPARE
030154 002344 WRIFOR :GET READY TO DO A WRIFOR
:WRITE HEADER AND DATA WITH 62 IN RHCS1

030156 004737 033064 JSR PC, @CHECKT :CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
:AND THAT ALL STATUS BITS ARE = 0
030162 104400 056662 TYPE ,CPHALT :CANNOT CONTINUE TESTING IF NOT
030166 000000 HALT :STOP TEST

030170 013777 004506 151770 MOV @RRP4VEC, @RP1EC :SET RPO4 VECTOR ADDRESS
:TO 'TIME1' IF P-CLOCK IS PRESENT
:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
: 'TIME' WILL ONLY SAVE
:CURRENT CYLINDER ADDRESS
:AND LOCK AHEAD REGISTERS

030176 013746 002344 MOV @WRIFOR, -(SP) :GET READY TO MOVE COMMAND
030202 052716 000101 BIS @GO!IE, (SP) :GET READY TO SET GO AND
:ENABLE INTERRUPT
030206 012677 151766 MOV (SP)+, @RHCS1 :GO WITH
:62 IN RHCS1 FOR WRITE HEADER AND DATA
:WITH INTERRUPT ENABLED
030212 011100 MOV @R1, R0 :SAVE RHCS1 DURING ABOVE OPERATION
030214 011305 MOV @R3, R5 :SAVE RHCS1 DURING ABOVE OPERATION

:*ONE REVOLUTION=16670 MICRO SEC. ONE SECTOR=760 MICRO SEC

030216 104412 WAT :WAIT FOR RDY BIT TO SET
030220 002200 RHCS1 :WAIT FOR RHCS1 REGISTER
030222 000200 RDY :WAIT FOR RDY BIT IN RHCS1 REGISTER
030224 003237 1695. :ALLOW 16950 MICRO SECONDS
030226 001515 845. :RDY MUST SET BETWEEN
:8500 AND 25400 MICRO SECONDS

:*NOW ONE MORE SECTOR HAS BEEN WRITTEN
:*'SC' WILL BE CHECKED TO MAKE SURE
:*NO ERRORS OCCURED

030230 017737 151744 002262 MOV @RHCS1, @RCS1 :GET RHCS1
030236 032737 100000 002262 BIT @SC, @RCS1 :IS 'SC' SET?
030244 001405 BEQ 35 :BRANCH IF "SPECIAL CONDITION" NOT SET
030246 004737 034726 JSR PC, @PLTREG :READ & SAVE ALL RH11 & RPO4 REGISTERS
030252 104072 ERROR 72 :THERE WAS AN UNDEFINED ERROR AFTER
:A WRITE HEADER AND DATA

030254 000137 030510 JMP TST36 : THIS IS A SETUP ERROR AND 'OPI' TEST CAN'T CONTI

```

;GO ON TO NEXT TEST

;*ONE SECTOR HAS BEEN FORMATTED NOW,
;*THE HARDWARE WILL BE CLEARED AND
;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.

030260 35:

030260 004737 033004

JSR PC,0#CLDISK

:SET R1-RHCS1, R2-RHCS2
:R3-RHDS1, R4-RHER1
:GIVE RH-11 INITIALIZE
:SETUP UNIT NUMBER

030274 013705 030126

MOV 0#15+12,R5

:GET SECTOR TRACK WORD

030274 005205 000026

INC R5

:+ 1

030274 122705 000026

CMPB #22.,R5

:IS IT 22 (WHOLE TRACK) ?

030274 001425 000026

BEG 45

:YES... DO NEXT TRACK

030300 010527 030126

MOV R5,0#15+12

:NO... RESTORE SECTOR TRACK FOR DATA WRITTEN

030304 010527 030142

MOV R5,0#25+6

:RESTORE SECTOR TRACK FOR "RUN" ROUTINE

030310 000410

BR 55

:CHECK FOR 46 SECTORS COMPLETED

030312 105037 030126

CLRB 0#15+12

:SET SECTOR = 0 FOR DATA WRITTEN

030316 105227 030127

INCB 0#15+13

:INCR TRACK FOR THE "RUN" ROUTINE

030322 105037 030142

CLRB 0#25+6

:SET SECTOR = 0 FOR DATA WRITTEN

030326 105227 030143

INCB 0#25+7

:INCR TRACK FOR THE "RUN" ROUTINE

030332 005337 004652

DEC 0#TMP1

:ARE 46 SECTORS DONE ?

030336 001266

BNE 15

:CONTINUE IF NOT

;*NOW 46 SECTORS HAVE BEEN FORMATTED

;*READ HEADER AND DATA FOR 46 SECTORS=11960 WORDS
;*WITH BUS ADDRESS INHIBITED

030340 004737 033004

JSR PC,0#CLDISK

:SET R1-RHCS1, R2-RHCS2
:R3-RHDS1, R4-RHER1
:GIVE RH-11 INITIALIZE
:SETUP UNIT NUMBER

;*FILL READ HEADER AND DATA COMMAND

030344 004027 034766

JSR RD,0#RUN

:SETUP TO RUN FOR DATA COMMAND

030350 000000

.BYTE 21.

:CYLINDER 0

030354 000000

.BYTE 0

:SECTOR 21.

030354 150510

-11956.-4

:TRACK 0
:WORD COUNT (DATA) = 11956. *

030356 003424

REINTC

:4 HEADER WORDS
:BUS ADDRESS

030360 000010

BAT

:STARTING ADDRESS OF DATA

030362 014000

FMT22!ECI

:BUFFER = RE:NTC
:INHIBIT BUS ADDRESS INCREMENT
:16 BITS PER WORD FORMAT



E02

MNDEC-11-DJRJJ-A, RFO4 5 6 FUNCT. CONT. TST-PT 2
DJRJJ.A.P11 T35

MAC111 27(655) 30-MAR-76 18:59 PAGE 175

SEQ 0223

OPERATION INCOMPLETE - RHER1 (BIT #13) OPI

8244
8245
8246 030510

78:

;INDEX PULSES, CAUSED AN UNDEFINED ERROR
;CONTINUE WITH THE NEXT TEST

78


```

0301 030602 012677 151372      MOV      (SP)+, @RHCS1      ;ENABLE INTERRUPT
                                ;GO WITH
                                ;6 IN RHCS1 FOR RECALIBRATE
                                ;WITH INTERRUPT ENABLED
0302 030606 104412              WAT                      ;WAIT FOR DRY BIT TO SET
0303 030610 002222              RHDS1                    ;WAIT FOR RHDS1 REGISTER
0304 030612 000300              DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
0305 030614 250650              25000.                  ;ALLOW 25000 MICRO SECONDS
0306 030616 060650              25000.                  ;DRY MUST SET BETWEEN
                                ;00 AND 50000 MICRO SECONDS
0307 030620 004737 033004      JSR      PC, @CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                ;R3-RHDS1, R4-RHER1
                                ;GIVE RH-11 INITIALIZE
                                ;SETUP UNIT NUMBER
0308 030624 15:                ;*FILL WRITE FROM BUFFER WITH HEADER
0309 030624 004037 032630      JSR      RO, @FLHEAD     ;SAVE HEADER DATA IN WRFROM
0310 030630 002370              WRFROM                  ;LOCATION WHERE SAVED
0311 030632 000004              4                       ;NUMBER OF WORDS SAVED
0312 030634 010000              10000                  ;FIRST DATA WORD
0313 030636 000000              <0*400>!0              ;SECOND DATA WORD
0314 030640 000000              0                       ;THIRD DATA WORD
0315 030642 000000              0                       ;FOURTH DATA WORD
0316 030644 25:                ;*FILL WRITE FROM BUFFER WITH DATA
0317 030644 004037 032654      JSR      RO, @CLAREA     ;CLEAR 256. WORDS. FROM WRFROM+10
0318 030650 002400              WRFROM+10              ;STARTING FROM WRFROM+10
0319 030652 000400              256.                   ;256. WORDS
0320 030654 000000              <0.*2000>!<0.*40>!0   ;FILL WITH <0.*2000>!<0.*40>!0
0321 030656 35:                ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
0322 030656 004037 034766      JSR      RO, @RUN        ;SETUP TO RUN FOR DATA COMMAND
0323 030662 000000              0                       ;CYLINDER 0
0324 030664 000              .BYTE 0                ;SECTOR 0
0325 030666 000              .BYTE 0                ;TRACK 0
0326 030668 177374              -256.-4                ;WORD COUNT (DATA) = 256. +
                                ;4 HEADER WORDS
0327 030670 002370              WRFROM                  ;BUS ADDRESS
                                ;STARTING ADDRESS OF DATA
                                ;BUFFER = WRFROM
0328 030672 000000              0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
0329 030674 010000              FMT22                  ;16 BITS PER WORD FORMAT

```

Handwritten mark

Handwritten mark

Handwritten mark

H02

030676
030677
030678
030679
030680
030681
030682
030683
030684
030685
030686
030687
030688
030689
030690
030691
030692
030693
030694
030695
030696
030697
030698
030699
030700
030701
030702
030703
030704
030705
030706
030707
030708
030709
030710
030711
030712
030713
030714
030715
030716
030717
030718
030719
030720
030721
030722
030723
030724
030725
030726
030727
030728
030729
030730
030731
030732
030733
030734
030735
030736
030737
030738
030739
030740
030741
030742
030743
030744
030745
030746
030747
030748
030749
030750
030751
030752
030753
030754
030755
030756
030757
030758
030759
030760
030761
030762
030763
030764
030765
030766
030767
030768
030769
030770
030771
030772
030773
030774
030775
030776
030777
030778
030779
030780

```

030676 002344 WRIFOR ;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A WRIFOR
;WRITE HEADER AND DATA WITH 62 IN RHCS1

030700 004737 033064 JSR PC,2#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
;AND THAT ALL STATUS BITS ARE = 0
030704 104400 056662 TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
030710 000000 HALT ;STOP TEST

030712 013777 004506 151246 MOV 2#RP4VEC,2#RVEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS

030720 013746 002344 MOV 2#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
030724 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT
030730 012677 151244 MOV (SP)+,2#RHCS1 ;GO WITH
;62 IN RHCS1 FOR WRITE HEADER AND DATA
;WITH INTERRUPT ENABLED

; *ONE REVOLUTION = 16670 MICRO SEC., ONE SECTOR = 760
; *MICRO SEC. MAX TIME ALLOWED = ONE REVOLUTION + HEAD
; *SWITCH + 2 SECTORS, MIN TIME ALLOWED = SECTOR (FIRST CASE)
; *IF THERE IS A FAILURE HERE HALT PROGRAM AFTER ERROR WITH
; *SWITCH 15 AND SEE CURRENT CYLINDER REGISTER TO DETERMINE
; *WHAT CYLINDER IS FAILING

030734 104412 WAT ;WAIT FOR RDY BIT TO SET
030736 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
030740 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
030742 003162 1650. ;ALLOW 16500 MICRO SECONDS
030744 001572 890. ;RDY MUST SET BETWEEN
;7600 AND 25400 MICRO SECONDS

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

; *SAVE REGISTERS IN SAVE TABLE
030746 004737 034726 JSR PC,2#PUTREG

030752 032737 040000 002304 BIT #ERR,2#DS1 ;ANY DISK ERRORS
030760 001004 BNE 9$ ;BRANCH IF YES
030762 032737 040000 002262 BIT #TRE,2#CS1 ;ANY RH ERRORS
030770 001401 BEQ 4$ ;BRANCH IF NO

030772 104066 9$: ERROR 66 ;SOME ERRORS OCCURRED
;WHILE DOING WRITE HEADER

```

Handwritten signature or initials

0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500

```

;AND DATA
;*THE FOLLOWING 3 ADDS SETS UP FOR NEXT TRACK WRITING
030774 062737 000400 030636 4$: ADD #400,2#1$+12 ;NEXT TRACK FOR HEADER
031002 062737 000040 030654 ADD #40,2#2$+10 ;NEXT TRACK FOR DATA
031010 062737 000400 030664 ADD #400,2#3$+6 ;NEXT TRACK FOR COMMAND

031016 005337 001200 DEC 2#STMP1 ;COUNT 19 TRACKS
031022 001300 BNE 1$

;*THE FOLLOWING CLEARS SETS UP FOR READ HEADER AND DATA
031024 005037 031104 CLR 2#SST3+12 ;START WITH SECTOR/TRACK = 0
031030 005037 031122 CLR 2#SST4+10 ;START WITH DATA = 0
031034 005037 031132 CLR 2#SST5+6 ;START WITH 0 FOR COMMAND

031040 004737 033004 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER

031044 012737 000023 001200 SST1: MOV #19.,2#STMP1 ;19 TRACKS TO BE READ

;*FILL READ INTO BUFFER WITH ALL ONES
031052 SST2:

031052 004037 032654 JSR RD,2#CLAREA ;CLEAR 260. WORDS, FROM REINTO
031056 003434 REINTO ;STARTING FROM REINTO
031060 000404 250. ;260. WORDS
031062 177777 -1. ;FILL WITH -1

031064 013737 031052 001110 MOV 2#SST2,2#SLPERR ;SET LOOP POINT

;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
031072 SST3:

031072 004037 032630 JSR RD,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
031076 002370 WRFROM ;LOCATION WHERE SAVED
031100 000004 4 ;NUMBER OF WORDS SAVED
031102 010000 10000 ;FIRST DATA WORD
031104 000000 0 ;SECOND DATA WORD
031106 000000 0 ;THIRD DATA WORD
031110 000000 0 ;FOURTH DATA WORD

;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
031112 SST4:

031112 004037 032654 JSR RD,2#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+(4*2)
031116 002400 WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
031120 000400 256. ;256. WORDS
031122 000000 <0.*2000>!<0*40>!0 ;FILL WITH <0.*2000>!<0*40>!0

```

29 M.

```

0463
0464
0465 031124          SSTS:  ;*FILL COMMAND FOR READ HEADER AND DATA
0466
0467 031124 004037 034756 JSR   RD, @#RUN      ;SETUP TO RUN FOR DATA COMMAND
0468 031130 000000          ;CYLINDER 0
0469 031132          .BYTE 0      ;SECTOR 0
0470 031133          .BYTE 0      ;TRACK 0
0471 031134 177374      -256.-4 ;WORD COUNT (DATA) = 256. +
0472          ;4 HEADER WORDS
0473 031136 003434      REINTO  ;BUS ADDRESS
0474          ;STARTING ADDRESS OF DATA
0475          ;BUFFER = REINTO
0476 031140 000000          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
0477 031142 014000      ECI!FMT22 ;16 BITS PER WORD FORMAT
0478          ;INHIBIT ECC CORRECTION
0479          ;DO NOT INHIBIT HEADER COMPARE
0480 031144 002350      REFOR   ;GET READY TO DO A REFOR
0481          ;READ HEADER AND DATA WITH 72 IN RHCS1
0482
0483
0484 031146 004737 033064 JSR   PC, @#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
0485          ;AND THAT ALL STATUS BITS ARE = 0
0486 031152 104400 056662 TYPE  .CPHALT      ;CANNOT CONTINUE TESTING IF NOT
0487 031156 000000      HALT   ;STOP TEST
0488
0489 031160 013777 004506 151000 MOV  @#RPHVEC, @#RVEC ;SET RPO4 VECTOR ADDRESS
0490          ;TO 'TIME1' IF P-CLOCK IS PRESENT
0491          ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
0492          ;'TIME' WILL ONLY SAVE
0493          ;CURRENT CYLINDER ADDRESS
0494          ;AND LOOK AHEAD REGISTERS
0495
0496
0497 031166 013746 002350 MOV  @#REFOR, -(SP) ;GET READY TO MOVE COMMAND
0498 031172 052716 000101 BIS  @#GO!IE, (SP)  ;GET READY TO SET GO AND
0499          ;ENABLE INTERRUPT
0500 031176 012677 150776 MOV  (SP)+, @#RHCS1 ;GO WITH
0501          ;72 IN RHCS1 FOR READ DATA
0502          ;WITH INTERRUPT ENABLED
0503
0504
0505 031202 104412          WAT           ;WAIT FOR RDY BIT TO SET
0506 031204 002200          RHCS1        ;WAIT FOR RHCS1 REGISTER
0507 031206 000200          RDY           ;WAIT FOR RDY BIT IN RHCS1 REGISTER
0508 031210 003162          1650.        ;ALLOW 16500 MICRO SECONDS
0509 031212 001572          890.          ;RDY MUST SET BETWEEN
0510          ;7600 AND 25400 MICRO SECONDS
0511
0512          ;*NOW SECTOR 0 OF ONE TRACK HAS BEEN READ CHECK COMPOSIT
0513          ;*ERROR BIT TO BE SURE NO ERROR HAPPENED
0514
0515          ;*SAVE REGISTERS IN SAVE TABLE
0516 031214 004737 034726 JSR   PC, @#PUTREG

```

```

0517
0518 031220 032737 040000 002304 BIT #ERR,2#DS1 ;ANY DISK ERRORS
0519 031226 001004 BNE 10$ ;BRANCH IF YES
0520 031230 032737 040000 002262 BIT #TRE,2#CS1 ;ANY RH ERRORS
0521 031236 001401 BEQ 11$ ;BRANCH IF NO
0522
0523 031240 104066 10$: ERROR 66 ;SOME ERRORS OCCURRED
0524 ;WHILE DOING READ
0525 ;HEADER AND DATA
0526
0527
0528 ;*NOW THE READ DATA WILL BE COMPARED DATA IN EACH SECTOR
0529 ;*IS UNIQUE IF PROGRAM IS HALTED ON ERROR THEN LOOK AT
0530 ;*RHDST TO GET WHAT TRACK IS IN ERROR. LOOKING AT THE DATA
0531 ;*BITS NO 4,5,6,7,8 IN GOOD DATA ALSO GIVES TRACK NUMBER
0532 ;*IN GOOD DATA ALSO GIVES TRACK NUMBER
0533
0534 031242 11$:
0535
0536 031242 004037 035032 JSR RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMCRY
0537 031246 002370 WRFROM ;GOOD DATA STARTS FROM WRFROM
0538 031250 003434 REINTO ;TEST DATA STARTS FROM REINTO
0539 031252 000404 260. ;260. WORDS TO BE COMPARED
0540 031254 031260 12$ ;RETURN TO 12$ ON ERROR
0541 031256 031264 13$ ;RETURN TO 13$ ON NO ERROR
0542
0543
0544 031260 104067 12$: ERROR 67 ;BITS 4,5,6,7,8
0545 031262 000207 RTS PC ;READ HEADER AND DATA
0546 ;ERROR
0547 ;HEAD SELECTION ERROR
0548 ;DATA READ GIVES NATURE
0549 ;OF ERROR
0550 ;EXCEPT FOR THE
0551 ;FOUR HEADER WORDS
0552 ;THE BITS 4,5,6,7,8
0553 ;GIVE THE TRACK NUMBER
0554
0555 ;*NOW INCREMENT TO READ NEXT TRACK
0556 031264 062737 000400 031104 13$: ADD #400,2#SST3+12 ;NEXT TRACK FOR HEADER
0557 031272 062737 000040 031122 ADD #40,2#SST4+10 ;NEXT TRACK FOR DATA
0558 031300 062737 000400 031132 ADD #400,2#SST5+6 ;NEXT TRACK FOR COMMAND
0559
0560 031306 005337 001200 DEC 2#STMP1 ;COUNT 19 TRACKS
0561 031312 001001 BNE 5$
0562
0563 031314 000402 BR TST37 ;TO NEXT TEST
0564
0565 031316 000137 031052 5$: JMP 2#SST2 ;JUMP BACK

```



```

000000 031416 002222          RHDS1          :WAIT FOR RHDS1 REGISTER
000001 031426 002222          DRY           :WAIT FOR DRY BIT IN RHDS1 REGISTER
000002 031426 060650          25000.        :ALLOW 25000 MICRO SECONDS
000003 031424 060650          25000.        :DRY MUST SET BETWEEN
000004                                     :00 AND 50000 MICRO SECONDS
000005
000006 ;*THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE MAX.
000007 ;*CYLINDER DIFFERENCE IS SET UP
000008
000009 ::*****
000010 031426 005737 004636      TST      0#RPO6 ;TEST FOR RPO6 DRIVE
000011 031432 001404          BEQ      14$   ;TREAT UNIT AS AN RPO4
000012                                     ;TREAT AS AN RPO6
000013 ::*****
000014
000015 031434 012737 001001 001200      MOV      #513.,0#STMP1 ;513 CYLINDERS
000016 031442 000403          BR       15$   ;CONTINUE
000017
000018 031444 012737 000401 001200      MOV      #257.,0#STMP1 ;257 CYLINDERS
000019 031452          14$:      ;CONTINUE WITH TEST
000020          15$:
000021
000022 031452 004737 003004          JSR      PC,0#CLDISK ;SET R1-RHCS1, R2-RHCS2
000023                                     ;R3-RHDS1, R4-RHER1
000024                                     ;GIVE RH-11 INITIALIZE
000025                                     ;SETUP UNIT NUMBER
000026
000027 ;*FILL WRITE FROM BUFFER WITH HEADER
000028 15:
000029
000030 031456          JSR      R0,0#FLHEAD ;SAVE HEADER DATA IN WRFROM
000031          WRFROM ;LOCATION WHERE SAVED
000032          4      ;NUMBER OF WORDS SAVED
000033          10000 ;FIRST DATA WORD
000034          0      ;SECOND DATA WORD
000035          0      ;THIRD DATA WORD
000036          0      ;FOURTH DATA WORD
000037
000038 ;*FILL WRITE FROM BUFFER WITH DATA
000039 25:
000040
000041 031476          JSR      R0,0#CLAREA ;CLEAR 256. WORDS. FROM WRFROM+10
000042          WRFROM+10 ;STARTING FROM WRFROM+10
000043          256. ;256. WORDS
000044          0      ;FILL WITH 0
000045
000046 ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
000047 35:
000048
000049 031510          JSR      R0,0#RUN ;SETUP TO RUN FOR DATA COMMAND
000050          0      ;CYLINDER 0
000051          .BYTE 0 ;SECTOR 0
000052          .BYTE 0 ;TRACK 0

```

```

8674 031520 177374 -256.-4 ;WORD COUNT (DATA) = 256. +
8675 ;4 HEADER WORDS
8676 031522 002370 WRFROM ;BUS ADDRESS
8677 ;STARTING ADDRESS OF DATA
8678 ;BUFFER = WRFROM
8679 031524 000000 0 ;DC NOT INHIBIT BUS ADDRESS INCREMENT
8680 031526 010000 FMT22 ;16 BITS PER WORD FORMAT
8681 ;DO NOT INHIBIT ECC CORRECTION
8682 ;DO NOT INHIBIT HEADER COMPARE
8683 031530 002344 WRIFOR ;GET READY TO DO A WRIFOR
8684 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
8685
8686
8687 031532 004737 033064 JSR PC,@#CHECKT ;CHECKS DVA, RDY, MOL, DPR, DRY AND VV = 1
8688 ;AND THAT ALL STATUS BITS ARE = 0
8689 031536 104400 056662. TYPE ,CPHALT ;CANNOT CONTINUE TESTING IF NOT
8690 031542 000000 HALT ;STOP TEST
8691
8692 031544 013777 004506 150414 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
8693 ;TO 'TIME1' IF P-CLOCK IS PRESENT
8694 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
8695 ;'TIME' WILL ONLY SAVE
8696 ;CURRENT CYLINDER ADDRESS
8697 ;AND LOOK AHEAD REGISTERS
8698
8699
8700 031552 013746 002344 MOV @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
8701 031556 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
8702 ;ENABLE INTERRUPT
8703 031562 012677 150412 MOV (SP)+,@RHCS1 ;GO WITH
8704 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
8705 ;WITH INTERRUPT ENABLED
8706
8707 ;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760
8708 ;*MICRO SECONDS, ONE SEEK = 7000 MICRO SECONDS.
8709 ;*MAX TIME = 1 REVOLUTION + 1 SEEK + 2 SECTORS
8710 ;*MIN TIME = 1 SECTOR
8711
8712
8713 031566 104412 WAT ;WAIT FOR RDY BIT TO SET
8714 031570 002200 RHCS1 ;WAIT FOR RHCS1 REGISTER
8715 031572 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8716 031574 002354 1260. ;ALLOW 12600 MICRO SECONDS
8717 031576 002354 1260. ;RDY MUST SET BETWEEN
8718 ;00 AND 25200 MICRO SECONDS
8719
8720 ;*NOW ONE SECTOR WRITE IS COMPLETE. CHANGES WILL BE MADE
8721 ;*FOR THE NEXT SECTOR, THEN THE ABOVE WILL BE REPEATED
8722 ;*UNTIL CYLINDER 256./512. IS REACHED
8723 031600 005237 031466 INC @#1$+10 ;CYLINDER HEADER DATA
8724 031604 005237 031506 INC @#2$+10 ;DATA
8725 031610 005237 031514 INC @#3$+4 ;CYLINDER COMMAND (RHCA)
8726 031614 005337 001200 DEC @#STMP1 ;COUNT DOWN FOR 256./512. CYLINDERS
8727 031620 001316 BNE 1$ ;DO NEXT WRITE IF 256./512. NOT DONE
  
```



```

031740 000000 0 :THIRD DATA WORD
031742 000000 0 :FOURTH DATA WORD
031744 65:
031744 004037 032654 JSR RO, @CLAREA :CLEAR 256. WORDS, FROM WRFROM+10
031750 002400 WRFROM+10 :STARTING FROM WRFROM+10
031752 000400 :256. WORDS
031754 000000 0 :FILL WITH 0

65:
031756 :*FILL READ HEADER AND DATA COMMAND
031756 004037 034766 JSR RO, @RUN :SETUP TO RUN FOR DATA COMMAND
031762 000000 0 :CYLINDER 0
031764 000 :SECTOR 0
031766 000 :TRACK 0
031768 177374 -256.-4 :WORD COUNT (DATA) = 256. +
031770 033434 REINTC :4 HEADER WORDS
:BUS ADDRESS
:STARTING ADDRESS OF DATA
:BUFFER = REINTC
:DO NOT INHIBIT BUS ADDRESS INCREMENT
031772 000000 0 :INHIBIT ECC CORRECTION
031774 014000 ECI!FMT22 :DO NOT INHIBIT HEADER COMPARE
:16 BITS PER WORD FORMAT
031776 002350 REFOR :GET READY TO DO A REFOR
:READ HEADER AND DATA WITH 72 IN RHCS!

032000 004737 033064 JSR PC, @CHECKT :CHECKS DVA, RDY, MCL, DPR, DRY AND VV = 1
:AND THAT ALL STATUS BITS ARE = 0
032004 104400 056662 TYPE .OPHALT :CANNOT CONTINUE TESTING IF NOT
032010 000000 HALT :STOP TEST
032012 013777 004506 150146 MOV @RPH4VEC, @RPHVEC :SET RPO4 VECTOR ADDRESS
:TO 'TIME1' IF P-CLOCK IS PRESENT
:OR TO 'TIME2' IF P-CLOCK IS NOT THERE
: 'TIME' WILL ONLY SAVE
:CURRENT CYLINDER ADDRESS
:AND LOCK AHEAD REGISTERS

:*ONE SECTOR = 760 MICRO SECONDS, ONE REVOLUTION =
:*16670 MICRO SECONDS, MAX SEEK = .52000 MICRO SECONDS
:*MAX TIME = ONE REV + 1 SEEK + 1 SECTOR
:*MIN TIME = 1 SECTOR

032020 013746 002350 MOV @REFOR, -(SP) :GET READY TO MOVE COMMAND
032024 052716 000101 BIS @GO!IE, (SP) :GET READY TO SET GO AND
:ENABLE INTERRUPT
032030 012677 150144 MOV (SP)+, @RHCS! :GO WITH
:72 IN RHCS! FOR READ DATA
:WITH INTERRUPT ENABLED

```


89370
89371
89372
89373
89374
89375
89376
89377
89378
89379
89380
89381
89382
89383
89384
89385
89386
89387
89388
89389
89390
89391
89392
89393
89394
89395
89396
89397
89398
89399
90000
90001
90002
90003
90004
90005
90006
90007
90008
90009
90010
90011
90012
90013
90014
90015
90016

032506
032506 000004
032510 005037 001102
032514 005037 001212
032520 005237 001100
032524 042737 100000 001100
032532 005327
032534 000001
032536 003022
032540 012737
032542 000001
032544 032534
032546 104400 032613
032552 013746 001100
032556 104404
032560 104400 032610
032564 013700 000042
032570 001405
032572 000905
032574 004710
032576 000240
032500 000240
032602 000240
032604
032604 000137
032606 006176
032610 377 000
032613 015 042412 042116
032620 050040 051501 020123
032626 000043

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

.SBTTL END OF PASS ROUTINE

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

\$EOP:
SCOPE
CLR \$TSTNM ;;ZERO THE TEST NUMBER
CLR \$TIMES ;;ZERO THE NUMBER OF ITERATIONS
INC \$PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,\$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
\$EOPCT: .WORD 1
BGT \$DOAGN ;;YES
MOV (PC)+,2(PC)+ ;;RESTORE COUNTER
\$ENDCT: .WORD 1
\$EOPCT
TYPE \$ENDMG ;;TYPE "END PASS #"
MOV \$PASS,-(SP) ;;SAVE \$PASS FOR TYPEOUT
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE \$ENULL ;;TYPE A NULL CHARACTER
\$GET42: MOV #42,R0 ;;GET MONITOR ADDRESS
BEG \$DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
\$ENDAD: JSR PC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11
\$DOAGN: JMP 2(PC)+ ;;RETURN
\$RTNAD: .WORD TST1
\$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
\$ENDMG: .ASCIZ '15'<'12'>'END PASS #'

9017
9018
9019
9020
9021
9022
9023
9024
9025
9026
9027
9028
9029
9030
9031
9032
9033
9034
9035
9036
9037
9038
9039
9040
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050
9051
9052
9053
9054
9055
9056
9057
9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
9068
9069
9070

032630
032630 010146
032632 010246
032634 012001
032636 012002

032640 012021
032642 005302
032644 001375
032646 012602
032650 012601
C32652 000200

032654
032654 010146
032656 010246
032660 010346
032662 012001
032664 012002
032666 012003
032670 010321
032672 005302
032674 001375
032676 012603

: THIS FILLS MEMORY WITH GIVEN DATA
: USED CHIEFLY FOR HEADER INFORMATION

: CALL IS
: JSR RD, 2#FLHEAD : FILL HEADER
: LOC : LOCATION WHERE SAVED
: XN : NUMBER OF WORDS
: XD1 : DATA REPEATED XN TIMES
: XD2 : DATA REPEATED XN TIMES

FLHEAD:
MOV R1, -(SP) : PUSH R1 ON STACK
MOV R2, -(SP) : PUSH R2 ON STACK
MOV (RD)+, R1 : R1 HAS ADDRESS OF WHERE TO SAVE
MOV (RD)+, R2 : R2 HAS NUMBER OF WORDS

: *NOW FILL DATA

IS: MOV (RD)+, (R1)+ : SAVE DATA
DEC R2 : DECREMENT COUNT
BNE IS : BRANCH IF INCOMPLETE
MOV (SP)+, R2 : POP STACK INTO R2
MOV (SP)+, R1 : POP STACK INTO R1
RTS RD

: THIS CLEARS ANY BLOCK OF MEMORY.
: FILLING IT WITH ANY DATA

: CALL IS
: JSR RD, 2#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

CLAREA:
MOV R1, -(SP) : PUSH R1 ON STACK
MOV R2, -(SP) : PUSH R2 ON STACK
MOV R3, -(SP) : PUSH R3 ON STACK
MOV (RD)+, R1 : FROM
MOV (RD)+, R2 : NUMBER
MOV (RD)+, R3 : DATA
IS: MOV R3, (R1)+ : MOVE DATA
DEC R2 : COUNT
BNE IS : BRANCH IF NOT COMPLETE
MOV (SP)+, R3 : POP STACK INTO R3

```

9071 032700 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
9072 032702 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
9073 032704 000200      RTS          RO        ;;RETURN TO MAIN PROGRAM

```

```

:THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
:WITH GIVEN VALUE
:CALL IS

```

```

:      JSR      RO,2*FILLRE
:      RHXX           ;REGISTER NAME
:      D            ;DATA

```

FILLRE:

```

MOV      R1,-(SP)      ;;PUSH R1 ON STACK
MOV      R2,-(SP)      ;;PUSH R2 ON STACK
MOV      (RO)+,R1      ;ADDRESS OF REGISTER
MOV      (RO)+,R2      ;DATA
SUB      #RHWC,R1      ;OFFSET
MOV      R2,SAVERE(R1) ;DATA IS MOVED IN
MOV      (SP)+,R2      ;;POP STACK INTO R2
MOV      (SP)+,R1      ;;POP STACK INTO R1
RTS          RO        ;;RETURN TO MAIN PROGRAM

```

```

9074 000000 000000
9075 000000 000000
9076 000000 000000
9077 000000 000000
9078 000000 000000
9079 000000 000000
9080 000000 000000
9081 000000 000000
9082 000000 000000
9083 000000 000000
9084 000000 000000
9085 000000 000000
9086 000000 000000
9087 000000 000000
9088 000000 000000
9089 000000 000000
9090 000000 000000
9091 000000 000000
9092 000000 000000
9093 000000 000000
9094 000000 000000
9095 000000 000000
9096 000000 000000
9097 000000 000000
9098 000000 000000
9099 000000 000000
9100 000000 000000

```

```

002172
004512

```

```

032706 010146
032708 010246
032710 012001
032712 012002
032714 012002
032716 162701
032718 010261
032720 012602
032722 012601
032724 000200

```

```

9101 : THIS SUBROUTINE SETS UP FOR SEARCH
9102 : CALL IS
9103 : JSR RO, @#SRCH
9104 : C ; CYLINDER
9105 : S ; SECTOR
9106 : T ; TRACK
9107 :
9108 SRCH: MOV (RO)+, @RHCA ; SET DESIRED CYLINDER ADDRESS
9109 : MOV (RO)+, @RHDS1 ; SET DESIRED SECTOR/TRACK ADDRESS
9110 : MOV @#SERCH, @RHCS1 ; GET READY FOR SEARCH
9111 : ; WITH 30 IN RHCS1
9112 :
9113 032734 012077 147252
9114 032740 012077 147240
9115 032744 013777 002334 147226
9116 :
9117 032752 000200
9118 :
9119 :
9120 :
9121 :
9122 :
9123 :
9124 :
9125 :
9126 :
9127 :
9128 :

```

```

9129 : THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
9130 : CALL IS
9131 : JSR RO, @#SEEKCY
9132 : C ; CYLINDER
9133 :
9134 :
9135 :
9136 :
9137 :
9138 :
9139 :
9140 :
9141 :
9142 :
9143 :
9144 :
9145 :
9146 :
9147 :
9148 :
9149 :
9150 :
9151 :
9152 :
9153 :
9154 :
9155 :
9156 :
9157 :
9158 :
9159 :
9160 :
9161 :
9162 :
9163 :
9164 :
9165 :
9166 :
9167 :
9168 :
9169 :
9170 :
9171 :
9172 :
9173 :
9174 :
9175 :
9176 :
9177 :
9178 :
9179 :
9180 :
9181 :
9182 :
9183 :
9184 :
9185 :
9186 :
9187 :
9188 :
9189 :
9190 :
9191 :
9192 :
9193 :
9194 :
9195 :
9196 :
9197 :
9198 :
9199 :
9200 :

```

```

9129
9130      ; THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
9131      ; CALL IS
9132      ;
9133      ; JSR      RO, @#OFSET
9134      ;          0          ; MICRO INCHES OFFSET
9135      032770 052077 147214      OFFSET: BIS      (RO)+, @RHOF      ; SET OFFSET REGISTER
9136      032774 013777 002354 147175      MOV      @#OFSETC, @RHCS1 ; MOV14 INTO RHCS1
9137      033002 000200                      RTS      RO          ; RETURN TO MAIN PROGRAM
9138
9139
9140      033004 013701 002200      CLDISK: MOV     @#RHCS1, R1      ; R1 WILL BE CONTROL AND STATUS1
9141      033010 013702 002176      MOV     @#RHCS2, R2      ; R2 WILL BE CONTROL AND STATUS2
9142      033014 013703 002222      MOV     @#RHDS1, R3     ; R3 WILL BE DISK STATUS REGISTER1
9143      033020 013704 002202      MOV     @#RHER1, R4     ; R4 WILL BE ERROR REGISTER #1
9144
9145      033024 012712 000040      MOV     #CLR, @R2       ; CLEAR ALL REG.
9146      033030 013712 004616      MOV     @#UNIT, @P2    ; REINSTATE UNIT NO.
9147      033034 005011                      CLR     @R1            ; CLEAR FUNCTION BITS
9148      033036 000207                      RTS      PC

```

9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200

:THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
:AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
:IT MAY CHECK VOLUME VALID (VV) IN RHDS1, DEPENDING ON ENTRY POINT

```

033040 000000      PCJSR: 0          ;PC OF JSR
033042 011637 033040 CHECK: MOV      (SP),2#PCJSR ;SAVE PC OF JSR+4
033046 062737 000004 033040 SUB      #4,2#PCJSR ;GET PC OF JSR
033054 011346      MOV      @R3,-(SP) ;GET RHDS1
033056 052716 000100 BIS      #VV,(SP) ;DONT CHECK VV BIT
033062 000406      BR       CHECKC ;GOTO COMMON CHECK ROUTINE

033064 011637 033040 CHECKT: MOV     (SP),2#PCJSR ;SAVE PC OF JSR+4
033070 062737 000004 033040 SUB     #4,2#PCJSR ;GET PC OF JSR
033076 011346      MOV     @R3,-(SP) ;GET RHDS1 & DO VV CHECK AT 3$

033100 011146      CHECKC: MOV     @R1,-(SP) ;GET CSI
033102 042716 173577 BIC     #173577,(SP) ;CLEAR UNWANTED BITS
033106 022726 004200 CMP     #DVA:RDY,(SP)+ ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                ;AND BE READY
033112 001403      BEQ     3$ ;BRANCH IF GOOD
033114 011137 001122 MOV     @R1,2#SBCADR ;BAD DATA REGISTER (RHCS1)
033120 104062      ERROR  62 ;RHCS1 DID NOT HAVE DEVICE
                                ;AVAILABLE RIGHT AT THE START
                                ;ALL OTHER BITS SHOULD BE 0

033122 042716 102000 3$: BIC     #ATA:LB1,(SP) ;CLEAR UNWANTED BITS
033126 022726 010700 CMP     #MOL:DPR:DRY:VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
033130 001404      BEQ     7$ ;BRANCH IF GOOD
033134 011137 001122 MOV     @R3,2#SBCADR ;BAD DATA IN REGISTER (RHDS1)
033140 104061      ERROR  61 ;RHDS1 HAS SOME BITS OTHER
                                ;THAN MOL, DRY, DPR, VV SET
                                ;ALL OTHER BITS SHOULD BE 0

033142 000207      RTS     PC ;RETURN TO TEST AND HALT

033144 062716 000006 7$: ADD     #6,(SP) ;ADJUST STACK TO JUMP OVER HALT IN TEST
033146 000207      RTS     PC ;RETURN TO TEST AND CONTINUE

```



```

033414 030425 BR 7S ;OUT
033416 017737 177614 001125 4S: MOV ;*NOW TIME AND TOLERANCE WILL BE CHECKED
033418 032737 000100 146546 BIT ;WAITRE, J%SBDDAT ;REGISTER CONTENTS FOR TYPEOUT
033420 031402 BEQ ;IE, J%RACS1 ;DID ANY INTERRUPT OCCUR
033422 104003 ERROR 3 ;BRANCH IF YES
;INTERRUPT DID NOT OCCUR EVEN
;AFTER ONE BNE AND ONE MOV
;OF THE WAITED ON BIT SETTING
033436 030414 BR 7S ;OUT
033440 160201 5S: SUB R2, R1 ;R1 NOW HAS LOWER LIMIT OF TIME
033442 023701 033242 CMP ;WAITTM, R1 ;FOR GOOD RESULTS, WAITTM
;MUST BE GREATER OR EQUAL
;TORI
033446 103002 BHS 6S ;BRANCH IF GOOD
033450 104004 ERROR 4 ;BIT DID OCCUR BUT TIME
;TAKEN IS BELOW LOWER LIMIT
033452 000406 BR 7S ;OUT
033454 060202 6S: ADD R2, R2 ;DOUBLE TOLERANCE
033456 060201 ADD R2, R1 ;R1 NOW HAS UPPER LIMIT OF TIME
033460 020137 033242 CMP R1, ;WAITTM ;FOR GOOD RESULTS, WAITTM
;MUST BE LESS OR EQUAL TO R1
033464 103001 BHS 7S ;BRANCH IF GOOD
033466 104004 ERROR 4 ;BIT DID OCCUR BUT TIME TAKEN
;IS ABOVE UPPER LIMIT
033470 7S: MOV (SP)+, R3 ;POP STACK INTO R3
033472 012603 MOV (SP)+, R2 ;POP STACK INTO R2
033474 012602 MOV (SP)+, R1 ;POP STACK INTO R1
033476 012601 MOV (SP)+, R0 ;POP STACK INTO R0
033500 RTI ;RETURN TO MAIN TEST

```

```

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

```

```

033502 177777 TIMCNT: 177777 ;COUNT FOR WAIT LOOP
033504 000025 RP*CTR: 25 ;COUNT FOR INTERRUPT WAIT (11 70 CPU)
033506 WAIT.T: MOV R0, - (SP) ;:PUSH R0 ON STACK
033508 010046

```



```

033510 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
033512 016600 000004      MOV      4(SP),R0      ;R0 HAS ADDRESS OF NEXT LOCATION
033514 010347 033224      MOV      R0,@WAITPC    ;WAT PC +2 IS IN WAITPC
033516 010347 033224      MOV      R0,@WAITPC    ;WAT PC IS IN WAITPC
033518 016237 033222 033234      SUB      #2,@WAITPC    ;WAT PC IS IN WAITPC
033520 010347 033226      MOV      @R0+,@WAITRE  ;WAIT ON REGISTER ADDRESS
033522 010347 033240      MOV      (R0)+,@WAITBT ;WAIT ON BIT
033524 010347 033240      CMP      (R0)+(R0)+    ;DUMP NEXT TWO WORDS-TA, TO
033526 010366 000004      MOV      R0,4(SP)     ;RESTORE RETURN ON STACK

; *THIS HAS THE TWO COUNT DOWNS FROM 177777

033546 010370 033502      MOV      @TIMCNT,R3    ;R3 HAS TEMPORARY COUNT
033548 032777 033240 177456 15:      BIT      @WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
033550 001025      BNE      4$           ;CHECK FOR THE INTERRUPT
033552 005303      DEC      R3           ;COUNT IF REQUIRED BIT NOT THERE
033554 001372      BNE      1$           ;
033556 010370 033502      MOV      @TIMCNT,R3    ;SECOND COUNT DOWN FROM 177777
033558 032777 033240 177436 25:      BIT      @WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
033560 001015      BNE      4$           ;CHECK FOR INTERRUPT
033562 005303      DEC      R3           ;COUNT IF REQUIRED BIT NOT THERE
033564 001372      BNE      2$           ;
033566 017737 177424 001126      MOV      @WAITRE,@$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
033568 032777 000100 146356      BIT      #IE,@RHCSI    ;DID ANY INTERRUPT OCCUR ?
033570 001402      BEQ      3$           ;BRANCH IF YES

033624 104001      ERROR    1            ;RPO4 DID NOT INTERRUPT
;BIT DID NOT OCCUR
033626 000417      R        5$           ;NUT -----)
033630 104002      3$:      ERROR    2            ;RPO4 INTERRUPTED BUT
;WAITED ON BIT DID NOT OCCUR
;EVEN AFTER TWO COUNT DOWNS
;FROM 177777 TO 0
033632 000415      BR       5$           ;OUT -----)

; *BIT DID SET SO CHECK IF INTERRUPT OCCURRED
; *THE AMOUNT OF TIME ALLOWED CAN BE CHANGED BY ALTERING LOCATION
; *"RPTCTR" ABOVE

033634 010370 033504      4$:      MOV      @RPTCTR,R3    ;LOAD COUNTER WITH COUNT
033640 005303      6$:      DEC      R3           ;COUNT DOWN ONE
033642 001376      BNE      6$           ;DO AGAIN IF NOT ZERO YET

033644 032777 000100 146326      BIT      #IE,@RHCSI    ;DID ANY INTERRUPT OCCUR ?
033652 001405      BEQ      5$           ;BRANCH IF YES
033654 017737 177356 001126      MOV      @WAITRE,@$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
033662 104003      ERROR    3            ;INTERRUPT DID NOT OCCUR
;EVEN AFTER ONE BNE OF
;THE WAITED ON BIT OCCURING
033664 000400      BR       5$           ;OUT -----)

```

00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000

SS:

MOV
MOV
RTI

SP+,R3
SP+,R0

::POP STACK INTO R3
::POP STACK INTO R0
:RETURN TO MAIN TEST


```

9456 033766 001374 BNE 1$ :BRANCH IF ALL NOT DONE
9457 033770 012604 MOV (SP)+,R4 :POP STACK INTO R4
9458 033772 012603 MOV (SP)+,R3 :POP STACK INTO R3
9459 033774 012602 MOV (SP)+,R2 :POP STACK INTO R2
9460 033776 012601 MOV (SP)+,R1 :POP STACK INTO R1
9461 034000 000200 RTS R0 :RETURN TO MAIN PROGRAM

```

```

:THIS IS A SUBROUTINE TO COMPARE REGISTERS
:GOOD DATA IS ALREADY SAVED IN 'SAVERE'
:TEST DATA IS IN THE REGISTERS
:CALL IS
:   JSR    R0,2#COMREG
:   SAVERE :GOOD DATA
:   RHCSI  :ADDRESS OF ADDRESS TEST DATA
:   N.     :RETURN FOR ERROR
:   RG     :RETURN FOR GOOD COMPARISON
:ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
:'REGADR' HAS REGISTER ADDRESS

```

COMREG:

```

9490 034002 010146 MOV R1,-(SP) :PUSH R1 ON STACK
9491 034002 010246 MOV R2,-(SP) :PUSH R2 ON STACK
9492 034004 010346 MOV R3,-(SP) :PUSH R3 ON STACK
9493 034006 010446 MOV R4,-(SP) :PUSH R4 ON STACK
9494 034010 010546 MOV R5,-(SP) :PUSH R5 ON STACK
9495 034012 012001 MOV (R0)+,R1 :R1 HAS ADDRESS OF GOOD DATA
9496 034014 012002 MOV (R0)+,R2 :R2 HAS ADDRESS OF ADDRESS OF TEST DATA
9497 034016 012003 MOV (R0)+,R3 :R3 HAS NUMBER OF WORDS
9498 034020 012004 MOV (R0)+,R4 :R4 HAS RETURN FOR ERROR
9499 034022 011000 MOV (R0),R0 :R0 HAS RETURN ON NO ERROR
9500 034024 011000
9501 :*NOW SAVE REGISTERS
9502 JSR PC,2#PUTREG :SAVE REGISTERS
9503 034032 113737 004537 002301 MOVB 2#SAVERE+25,2#AS+1:MAKE UPPER BYTE OF R HAS SAME
9504 034040 012705 177776 MOV #-2,R5 :PRESET R5 TO -2
9505 :*NOW COMPARES WILL MADE
9506 034044 062705 000002 1$: ADD #2,R5 :INCREMENT TO INDEX
9507 034050 022122 CMP (R1)+,(R2)+ :COMPARE REGISTER CONTENTS
9508 034052 001420 BEQ 2$ :BRANCH IF GOOD
9509 034054 014137 001124 MOV -(R1),2#$GDDAT :SAVE GOOD DATA
9510 034060 014237 001126 MOV -(R2),2#$BDDAT :SAVE BAD DATA
9511 034054 016537 002172 004500 MOV RHC(R5),2#REGADR :SAVE ADDRESS OF FAILING REGISTER
9512 034072 004714 JSR PC,2#R4 :RETURN TO MAIN PROGRAM
9513 :TO PRINT ERROR
9514 034074 022122 CMP (R1)+,(R2)+ :UNDO -(R1) AND -(R2) FOR ERRORS
9515 034076 017746 145036 MOV 2#SWR,-(SP) :GET SWITCH SETTING
9516 034102 042716 177177 BIC #10600,(SP) :KEEP ONLY SWITCH 7 AND 8
9517 034106 022726 000200 CMP #SW07,(SP)+ :IS 7 SET AND 8 DOWN
9518 034112 001402 BEQ 3$ :BRANCH OUT IF YES
9519 034114 005303 2$: DEC R3 :ARE ALL COMPARES DONE

```

```

9520 034116 001352          BNE      1$          ;BRANCH IF NOT COMPLETE
9521
9522 034120          3$:
9523 034120 012605          MOV      (SP)+,R5          ;;POP STACK INTO R5
9524 034122 012604          MOV      (SP)+,R4          ;;POP STACK INTO R4
9525 034124 012603          MOV      (SP)+,R3          ;;POP STACK INTO R3
9526 034126 012602          MOV      (SP)+,R2          ;;POP STACK INTO R2
9527 034130 012601          MOV      (SP)+,R1          ;;POP STACK INTO R1
9528 034132 000200          RTS      R0              ;RETURN TO MAIN PROGRAM
9529 034134 000000          4$:      .WORD      0          ;TEMP STORAGE

```

9530
9531
9532
9533
9534
9535
9536
9537
9538
9539
9540
9541
9542
9543
9544
9545
9546
9547
9548
9549
9550
9551
9552
9553
9554
9555
9556
9557
9558
9559
9560
9561
9562
9563
9564
9565
9566
9567
9568
9569
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579
9580
9581
9582
9583

034136 000000
034140
034140 005037 177776
034144 012737 177777 040434
034152 104400 034160
034156 000421

034222
034222 013746 004504
034226 104401
034230 104400 034236
034234 000414

034266
034266 013746 001110
034272 104401
034274 104400 001223
034280 104400 034306
034304 000430

034366
034366 104400 034374
034372 000420

034434
034434 104400 034442
034440 000432

034526
034526 104411
034530 062716 000002
034534 012637 001106
034540 104400 034546

:HERE IS A DETAILED EXPLANATION 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.

: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.

TESTAD: 0 ;FIRST ADDRESS OF TEST
OPERSEL:
CLR PS ;MAKE PROCESSOR STATUS ZERO
MOV #-1, @#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
TYPE ,65\$;TYPE ASCIZ STRING
BR ,64\$;GET OVER THE ASCIZ
;;65\$: .ASCIZ <15><12>/THE PROGRAM WAS IN TEST NUMBER /
64\$:
MOV @#TSTNM, -(SP) ;GET READY TO TYPE TEST
TYPOC ;NUMBER
TYPE ,67\$;TYPE ASCIZ STRING
BR ,66\$;GET OVER THE ASCIZ
;;67\$: .ASCIZ <15><12>/THE LOOP BACK PC WAS /
66\$:
MOV @#SLPERR, -(SP) ;GET READY TO TYPE LOOP BACK PC
TYPOC
TYPE , \$SCLF
TYPE ,69\$;TYPE ASCIZ STRING
BR ,68\$;GET OVER THE ASCIZ
;;69\$: .ASCIZ <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
68\$:
TYPE ,71\$;TYPE ASCIZ STRING
BR ,70\$;GET OVER THE ASCIZ
;;71\$: .ASCIZ <15><12>/TYPE THE FIRST PC OF THE TEST/
70\$:
TYPE ,73\$;TYPE ASCIZ STRING
BR ,72\$;GET OVER THE ASCIZ
;;73\$: .ASCIZ <15><12>/TO BE LOOPED ON FOLLOWED BY A CARRIAGE RETURN <15><12>
72\$:
RDOCT
ADD #2, (SP) ;GET LPADR
MOV (SP)+, @#SLPADR
TYPE ,75\$;TYPE ASCIZ STRING


```

9584 034544 000417
9585
9586 034604
9587 034604 104400 034612
9588 034610 000440
9589
9590 034712
9591 034712 104411
9592 034714 012637 001110
9593 034720 013746 001106
9594 034724 000002
9595
9596
9597
9598
9599
9600
9601
9602
9603
9604
9605
9606
9607
9608
9609
9610 034726
9611 034726 010046
9612 034730 010146
9613 034732 010246
9614 034734 012700 002172
9615 034740 012701 002254
9616 034744 012702 000022
9617 034750 013021
9618 034752 005302
9619 034754 001375
9620 034756 012602
9621 034760 012601
9622 034762 012600
9623 034764 000207

```

```

745: BR 745 ;:GET OVER THE ASCIZ
;:755: .ASCIZ <15><12>/TYPE THE PC WHERE YOU WANT/
746:
775: TYPE 775 ;:TYPE ASCIZ STRING
;:785: BR 785 ;:GET OVER THE ASCIZ
;:795: .ASCIZ <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN //15
796:
RDOCT
MOV (SP)+,2*SLPERR ;GET LPERR
MOV 2*SLPADR,-(SP)
RTI

```

```

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

```

PUTREG:

```

MOV R0,-(SP) ;:PUSH R0 ON STACK
MOV R1,-(SP) ;:PUSH R1 ON STACK
MOV R2,-(SP) ;:PUSH R2 ON STACK
MOV #RHC,R0 ;:STARTING ADDRESS OF REGISTERS
MOV #WC,R1 ;:STARTING ADDRESS OF SAVING LOCATIONS
MOV #RHC-RHC+2/2,R2 ;:NUMBER OF REG. INTO R2
105: MOV 2(R0)+,(R1)+ ;SAVE HARDWARE REG.
DEC R2
BNE 105
MOV (SP)+,R2 ;:POP STACK INTO R2
MOV (SP)+,R1 ;:POP STACK INTO R1
MOV (SP)+,R0 ;:POP STACK INTO R0
PC

```

```

9624
9625
9626
9627
9628
9629
9630
9631
9632
9633
9634
9635
9636
9637 034766 012077 145220
9638 034772 012077 145206
9639 034775 012077 145170
9640 035002 012077 145166
9641 035006 013746 004616
9642 035012 052016
9643 035014 012677 145156
9644 035020 012077 145164
9645
9646 035024 013077 145150
9647 035030 000200
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668 035032
9669 035032 010146
9670 035034 010246
9671 035036 010346
9672 035040 010446
9673 035042 010546
9674 035044 012001
9675 035046 012002
9676 035050 012003
9677 035052 012005

```

```

:THIS IS A DATA COMMAND SETUP SUBROUTINE
:THE CALL IS
:      JSR      RO, @#RUN
:      C
:      S
:      T
:      -W
:      B
:      BAI
:      FMT22!ECI!HCI
:
:      COM
:RUN:  MOV      (RO)+, @RHCA
:      MOV      (RO)+, @RHDS
:      MOV      (RO)+, @RHWC
:      MOV      (RO)+, @RHBA
:      MOV      @#UNIT, -(SP)
:      BIS      (RO)+, (SP)
:      MOV      (SP)+, @RHCS2
:      MOV      (RO)+, @RHOF
:
:      MOV      @ (RO)+, @RHCS1
:      RTS      RO
:
:CYLINDER
:SECTOR
:TRACK
:WORD COUNT
:BUS ADDRESS
:BUS ADDRESS INHIBIT
:FMT22=1 =16 BIT WORDS
:ECI = ECC CORRECTION INHIBIT
:HCI = HEADER COMPARE INHIBIT
:COMMAND ADDRESS
:CYLINDER
:DESIRED SECTOR/TRACK
:WORD COUNT
:BUS ADDRESS
:GET UNIT NO
:SET BUS ADDRESS INHIBIT
:UNIT NO AND BAI TO RHCS2
:FORMAT, ECC INHIBIT, HEADER
:COMPARE, IF THERE
:COMMAND IN RHCS1
:RETURN TO MAIN PROGRAM

```

```

:THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
: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
:CALL IS
:      JSR      RO, @#COMPAR
:      G
:      T
:      N
:      RE
:      RG
:
:      ADDRESS OF GOOD DATA
:      ADDRESS OF TEST DATA
:      NUMBER OF WORDS TO BE COMPARED
:      RETURN ON ERROR
:      RETURN ON NO ERROR

```

```

:COMPAR:
:      MOV      R1, -(SP)
:      MOV      R2, -(SP)
:      MOV      R3, -(SP)
:      MOV      R4, -(SP)
:      MOV      R5, -(SP)
:      MOV      (RO)+, R1
:      MOV      (RO)+, R2
:      MOV      (RO)+, R3
:      MOV      (RO)+, R5
:
:      PUSH R1 ON STACK
:      PUSH R2 ON STACK
:      PUSH R3 ON STACK
:      PUSH R4 ON STACK
:      PUSH R5 ON STACK
:      ADDRESS OF GOOD DATA BUFFER
:      ADDRESS OF TEST DATA BUFFER
:      NO OF WORDS TO BE COMPARED
:      RETURN ON ERROR

```

```

9678 035054 011000      MOV      (R0),R0      ;RETURN ON NO ERROR
9679 035056 010304      MOV      R3,R4      ;NO OF WORDS TO BE COMPARED
9680 035060 005204      INC      R4
9681 035062 010437 004502  1$:      MOV      R4,2*ERWORD ;FOR ERROR WORD NO
9682 035066 022122      CMP      (R1)+,(R2)+ ;COMPARE GOOD WITH TEST DATA
9683 035070 001417      BEQ      2$          ;BRANCH IF GOOD
9684
9685 035072 014137 001124      MOV      -(R1),2*$GDDAT ;GOOD DATA
9686 035076 014237 001126      MOV      -(R2),2*$BDDAT ;BAD DATA
9687 035102 160337 004502      SUB      R3,2*ERWORD ;ERROR WORD NO.
9688 035106 004715      JSR      PC,2R5 ;RETURN TO PRINT ERROR
9689 035110 022122      CMP      (R1)+,(R2)+ ;UNDO -(R1) AND -(R2) FOR ERRORS
9690 035112 017746 144022      MOV      2SWR,-(SP) ;GET SWITCH SETTING
9691 035116 042716 177177      BIC      #1C600,(SP) ;KEEP ONLY SWITCH 7 AND 8
9692 035122 022726 000200      CMP      #SW07,(SP)+ ;IS 7 SET AND 8 RESET
9693 035126 001402      BEQ      3$          ;BRANCH OUT IF YES
9694 035130 005303      2$:      DEC      R3          ;COUNT
9695 035132 001353      BNE      1$          ;BRANCH IF ALL NOT DEVICE
9696 035134      3$:
9697 035134 012605      MOV      (SP)+,R5    ;;POP STACK INTO R5
9698 035136 012604      MOV      (SP)+,R4    ;;POP STACK INTO R4
9699 035140 012603      MOV      (SP)+,R3    ;;POP STACK INTO R3
9700 035142 012602      MOV      (SP)+,R2    ;;POP STACK INTO R2
9701 035144 012601      MOV      (SP)+,R1    ;;POP STACK INTO R1
9702 035146 000200      RTS      R0          ;RETURN TO MAIN PROGRAM
9703      ;* THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
9704      ;* ADDRESS FROM 176700 TO ANY TYPED VALUE
9705
9706 035150      BASECH:
9707 035150 104400 035156      TYPE      65$      ;;TYPE ASCIZ STRING
9708 035154 000425      BR        64$      ;;GET OVER THE ASCIZ
9709      ;;65$: .ASCIZ <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS
9710 035230      64$:
9711 035230 013746 002200      MOV      2*RHCS1,-(SP) ;GET READY TO TYPE OLD BASE
9712 035234 104401      TYP0C
9713 035236 104400 035244      TYPE      67$      ;;TYPE ASCIZ STRING
9714 035242 000425      BR        66$      ;;GET OVER THE ASCIZ
9715      ;;67$: .ASCIZ <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'
9716 035316      66$:
9717 035316 004737 037026      JSR      PC,2*STKINT ;INITIALIZE THE TTY KEYBOARD
9718 035322 104411      RDOCT
9719 035324 012700 002170      MOV      #RHDB,R0    ;GET STARTING ADDRESS OF REGISTERS
9720 035330 012701 000024      MOV      #20,R1     ;NUMBER OF REGISTERS
9721 035334 042710 177700  1$:      BIC      #1C77,(R0) ;CLEAR OLD BASE
9722 035340 051620      BIS      (SP),(R0)+ ;SET NEW BASE
9723 035342 005301      DEC      R1          ;COUNT
9724 035344 001373      BNE      1$          ;BRANCH IF 20 NOT DONE
9725 035346 104400 035354      TYPE      69$      ;;TYPE ASCIZ STRING
9726 035352 000417      BR        68$      ;;GET OVER THE ASCIZ
9727      ;;69$: .ASCIZ <15><12>/PRESENT VECTOR ADDRESS IS
9728 035412      68$:
9729 035412 013746 002166      MOV      2*RPVEC,-(SP) ;GET READY TO TYPE OLD VECTOR ADDRESS
9730 035416 104401      TYP0C
9731 035420 104400 035426      TYPE      ,71$     ;;TYPE ASCIZ STRING

```

M04

MNDEC-11-DZRJJ-A, RFD4 5 6 FUNCT. CONT. TST-PT 2
 DZRJJ.A.P11 END OF PASS ROUTINE

MACY11 27(655) 30-MAR-76 18:59 PAGE 209

SEQ 0257

```

9732 035424 000437          BR      70$          ;;GET OVER THE ASCIZ
9733          ;;71$: .ASCIZ <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
9734 035524          70$:          RDOCT
9735 035524 104411          MOV     (SP)+,2*RPVEC  ;;SETUP VECTOR ADDRESS
9736 035526 012637 002166    TYPE   73$          ;;TYPE ASCIZ STRING
9737 035532 104400 035540    BR     72$          ;;GET OVER THE ASCIZ
9738 035536 000421          ;;73$: .ASCIZ <15><12>/RESTART PROGRAM FROM 200 OR 210/
9739          72$:          TYPE   75$          ;;TYPE ASCIZ STRING
9740 035602          BR     74$          ;;GET OVER THE ASCIZ
9741 035602 104400 035610    ;;75$: .ASCIZ <15><12>/NEW BASE WILL REMAIN - /
9742 035606 000416          74$:          MOV     2*RHCS1,-(SP)
9743          TYP0C
9744 035644 013746 002200    TYPE   77$          ;;TYPE ASCIZ STRING
9745 035644 104401          BR     76$          ;;GET OVER THE ASCIZ
9746 035650 104401          ;;77$: .ASCIZ <15><12>/NEW VECTOR WILL REMAIN - /
9747 035652 104400 035660    76$:          MOV     2*RPVEC,-(SP)
9748 035656 000416          TYP0C
9749          TYPE   79$          ;;TYPE ASCIZ STRING
9750 035714          BR     78$          ;;GET OVER THE ASCIZ
9751 035714 013746 002166    ;;79$: .ASCIZ <15><12>/ /
9752 035720 104401          78$:          TYPE   81$          ;;TYPE ASCIZ STRING
9753 035722 104400 035730    BR     80$          ;;GET OVER THE ASCIZ
9754 035726 000402          ;;81$: .ASCIZ <15><12>/UNTIL PROGRAM IS RELOADED/
9755          80$:          HALT
9756 035734          80$:
9757 035734 104400 035742    ;;81$:
9758 035740 000416          80$:
9759          ;;81$:
9760 035776          80$:
9761 035776 000000          80$:
9762
  
```

97763
97764
97765
97766
97767
97768
97769
97770
97771
97772
97773
97774
97775
97776
97777
97778
97779
97780

```
036000  
036000 104400 036006  
036004 000424  
  
036056  
036056 104401  
036060 012777 036000 144100  
036066 000000
```

```
::*****  
RPVECT:  
      TYPE    655                    ::TYPE ASCIZ STRING  
      BR      648                    ::GET OVER THE ASCIZ  
:655: .ASCIZ /UNEXPECTED INTERRUPT FROM RPO4 @ PC = /  
648:     
      TYPOC                        :TYPE FROM PC  
      MOV     @RPVECT,@RPVEC       :RESTORE TRAP RPO4 VECTOR  
      HALT                         :CHANGE TO CONTINUE  
:*****
```

.SBTTL SYSTEM LIBRARY ROUTINES

.SBTTL SCOPE HANDLER ROUTINE

 *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 *AND LOAD THE TEST NUMBER(SYSTNM) INTO THE DISPLAY REG. DISPLAY 7:0
 *AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY 15:08
 *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 *SW14=1 LOOP ON TEST
 *SW14=0 INHIBIT ITERATIONS
 *SW10=1 LOOP ON ERROR
 *SW10=0 LOOP ON TEST IN SWR 7:0
 *CALL SCOPE ::SCOPE=107

036100	001111	040000	143040	15:	BIT	#BIT14, \$SWR	:: TEST FOR CHANGE IN SOFT-SWR
036102	000416			16:	BNE	\$OVER	:: LOOP ON PRESENT TEST?
							:: YES IF SW14=1
036104	013746	000004			MOV	\$ERRVEC, -(SP)	:: IF RUNNING ON THE "XOR" TESTER CHANGE
036110	012737	036130	000004		MOV	\$S, \$ERRVEC	:: THIS INSTRUCTION TO A "NOP" (NOP=24C)
036116	005737	177060			TST	\$BIT7, 060	:: SAVE THE CONTENTS OF THE ERROR VECTOR
036122	012637	000004			MOV	(SP)+, \$ERRVEC	:: SET FOR TIMEOUT
036126	000463				BR	\$SVLAD	:: TIME OUT ON XOR?
036130	022626			55:	CMR	(SP)+, (SP)+	:: RESTORE THE ERROR VECTOR
036132	012637	000004			MOV	(SP)+, \$ERRVEC	:: GO TO THE NEXT TEST
036136	000423				BR	\$S	:: CLEAR THE STACK AFTER A TIME OUT
036140	032777	000400	142772	65: ::	END OF	CODE FOR THE XOR	:: RESTORE THE ERROR VECTOR
036146	001404				BIT	#BIT08, \$SWR	:: LOOP ON THE PRESENT TEST
036150	127737	142764	001102		BNE	\$S	:: LOOP ON SPEC. TEST?
036156	001462				CMR	\$SWR, \$YSTNM	:: BR IF NO
036160	105737	001103		25:	BNE	\$OVER	:: ON THE RIGHT TEST? SWR(7:0)
036164	001421				BNE	\$SERFLG	:: BR IF YES
036166	123737	001115	001103		CMR	\$SERMAX, \$SERFLG	:: HAS AN ERROR OCCURRED?
036174	101015				BHI	\$S	:: BR IF NO
036176	032777	001000	142734		BIT	#BIT09, \$SWR	:: MAX. ERRORS FOR THIS TEST OCCURED"
036204	001404				BNE	\$S	:: BR IF NO
036206	013737	001110	001106	75:	MOV	\$LPERR, \$LPADR	:: LOOP ON ERROR?
036214	000443				BR	\$OVER	:: BR IF NO
036216	105037	001103		45:	CLRB	\$SERFLG	:: SET LOOP ADDRESS TO LAST SCOPE
036222	005037	001212			CLR	\$STIMES	:: ZERO THE ERROR FLAG
036226	000415				BR	\$S	:: CLEAR THE NUMBER OF ITERATIONS TO MAKE
036230	032777	004000	142702	35:	BIT	#BIT11, \$SWR	:: ESCAPE TO THE NEXT TEST
036236	001011				BNE	\$S	:: INHIBIT ITERATIONS?
036240	005737	001100			TST	\$PASS	:: BR IF YES
036244	001406				BNE	\$S	:: IF FIRST PASS OF PROGRAM
036246	005237	001104			INC	\$ICNT	:: INHIBIT ITERATIONS
036252	023737	001212	001104		CMR	\$STIMES, \$ICNT	:: INCREMENT ITERATION COUNT
							:: CHECK THE NUMBER OF ITERATIONS MADE

000000	036360	002021
000001	036362	012737
000002	036364	013737
000003	036366	105237
000004	036370	011637
000005	036372	011637
000006	036374	005037
000007	036376	112737
000008	036378	013737
000009	036380	013737
000010	036382	000037
000011	036384	000037
000012	036386	000037
000013	036388	000037
000014	036390	000037
000015	036392	000037
000016	036394	000037
000017	036396	000037
000018	036398	000037
000019	036400	000037

000001	001104	IS:
036340	001212	
001102		SSVLAD:
001106		
001110		
001214		
000001	001115	SOVER:
001102	142610	
001106		
		SMXONT: 4

SOVER	BGE
#1, SCONT	MOV
SMXONT, STIMES	MOV
STSTNM	INCB
(SP), \$LPADR	MOV
(SP), \$LPERR	MOV
\$ESCAPE	CLR
#1, \$ERMAX	MOVB
SYSTEM, \$DISPLAY	MOV
\$LPADR, (SP)	RTI

```

::BR IF MORE ITERATION REQUIRED
::REINITIALIZE THE ITERATION COUNTER
::SET NUMBER OF ITERATIONS TO DO
::COUNT TEST NUMBERS
::SAVE SCOPE LOOP ADDRESS
::SAVE ERROR LOOP ADDRESS
::CLEAR THE ESCAPE FROM ERROR ADDRESS
::ONLY ALLOW ONE(1) ERROR ON NEXT TEST
::DISPLAY TEST NUMBER
::PUSH RETURN ADDRESS
::PUSH PS
::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
* TYPOS ;:GO TO THE ROUTINE

\$TYPOS:

MOV R0, -(SP) ;:PLSH 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(SF), R5 ;:GET THE INPUT NUMBER
BPL R5 ;:BR IF INPUT IS POS.
NEG R5 ;:MAKE THE BINARY NUMBER POS.
MOVB #'-(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
MOV \$DTBL(R0), R1 ;:GET THE CONSTANT
3\$: SUB R1, R5 ;:FORM THIS BCD DIGIT
BLT R5 ;:BR IF DONE
INC R2 ;:INCREASE THE BCD DIGIT BY 1
4\$: ADD R1, R5 ;:ADD BACK THE CONSTANT
TST R2 ;:CHECK IF BCD DIGIT=0
BYE R5 ;:FALL THROUGH IF 0
TSTB (SP) ;:STILL DOING LEADING 0'S?
BMI R5 ;:BR IF YES
5\$: ASLB (SP) ;:MSD?
BCC R5 ;:BR IF NO
MOVB 1(SP), -(R3) ;:YES--SET THE SIGN
6\$: BIS #'C, R2 ;:MAKE THE BCD DIGIT ASCII
7\$: BIS #'(R2) ;:MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB R2, (R3)+ ;:PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST (R0)+ ;:JUST INCREMENTING
CMP R0, #10 ;:CHECK THE TABLE INDEX
BLT R5 ;:GO DO THE NEXT DIGIT
BGT R5 ;:GO TO EXIT
MOV R5, R2 ;:GET THE LSD
BR R5 ;:GO CHANGE TO ASCII
8\$: TSTB (SP)+ ;:WAS THE LSD THE FIRST NON-ZERO?
BPL R5 ;:BR IF NO
MOVB -(SP), -(R3) ;:YES--SET THE SIGN FOR TYPING
9\$: CLRB (R3) ;:SET THE TERMINATOR
MOV (SP)+, R5 ;:POP STACK INTO R5

036342 010046
036344 010146
036346 010246
036350 010346
036352 010546
036354 012746 020200
036360 016605 000020
036364 100004
036366 005405
036370 112766 000055 000001
036376 005200 1\$:
036400 012703 036556
036404 112723 000040
036410 005002 2\$:
036412 016001 036546
036416 160105 3\$:
036420 002402
036422 005202
036424 000774
036426 060105 4\$:
036430 005702
036432 001002
036434 105716
036436 100407
036440 106316 5\$:
036442 103003
036444 116663 000001 177777
036452 052702 000060 6\$:
036456 052702 000040 7\$:
036462 110223
036464 005720
036466 020027 000010
036472 002746
036474 003002
036476 010502
036500 000764
036502 105726 8\$:
036504 100003
036506 116663 177777 177776
036514 105013 9\$:
036516 012605

DZJJA.P11

CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

99999	036520	012603
99999	036522	012602
99999	036524	012601
99999	036526	012600
99999	036530	104400
99999	036534	016666
99999	036542	012656
99999	036544	000002
99999	036546	023420
99999	036550	001750
99999	036552	000144
99999	036554	000012
99999	036556	000004

036556 000002 000004

SDTBL: 10000.

1000.

100.

10.

SDBLK: .BLKW 4

```

MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
TYPE $DBLK
MOV 2(SP),4(SP)
MOV (SP)+,(SP)

```

```

::POP STACK INTO R3
::POP STACK INTO R2
::POP STACK INTO R1
::POP STACK INTO R0
::NOW TYPE THE NUMBER
::ADJUST THE STACK
::RETURN TO USER

```

.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
*

```

```

9900
9901
9902
9903
9904
9905
9906
9907
9908
9909
9910
9911
9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930
9931
9932
9933
9934
9935
9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961

```

```

$TYPE: TSTB $TPFLG ::IS THERE A TERMINAL?
BPL 1$ ::BR IF YES
HALT ::HALT HERE IF NO TERMINAL
BR 3$ ::LEAVE
1$: MOV RO,-(SP) ::SAVE RO
MOV 2$(SP),RO ::GET ADDRESS OF ASCIZ STRING
2$: MOVB (RO)+,-(SP) ::PUSH CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ::BR IF IT ISN'T THE TERMINATOR
TST (SP)+ ::IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+,RO ::RESTORE RO
3$: ADD #2,(SP) ::ADJUST RETURN PC
RTI ::RETURN
4$: CMPB #HT,(SP) ::BRANCH IF <HT>
BEQ 8$
CMPB #CRLF,(SP) ::BRANCH IF NOT <CRLF>
BNE 5$
TST (SP)+ ::POP <CR><LF> EQUIV
TYPE A CR AND LF
37002: CLRB $CHARCNT ::CLEAR CHARACTER COUNT
BR 2$ ::GET NEXT CHARACTER
5$: JSR PC,$TYPEC ::GO TYPE THIS CHARACTER
6$: CMPB $FILLC,(SP)+ ::IS IT TIME FOR FILLER CHARS.?
BNE 2$ ::IF NO GO GET NEXT CHAR.
MOV $NULL,-(SP) ::GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
7$: DECB 1$(SP) ::DOES A NULL NEED TO BE TYPED?
BLT 6$ ::BR IF NO--GO POP THE NULL OFF OF STACK
JSR PC,$TYPEC ::GO TYPE A NULL
DECB $CHARCNT ::DO NOT COUNT AS A COUNT
BR 7$ ::LOOP

```

:HORIZONTAL TAB PROCESSOR

```

8$: MOVB #'(SP) ::REPLACE TAB WITH SPACE
9$: JSR PC,$TYPEC ::TYPE A SPACE
BITB #7,$CHARCNT ::BRANCH IF NOT AT

```

036730	001372			BNE	9\$:: TAB STOP
036732	005726			TST	(SP)+	:: POP SPACE OFF STACK
036734	000724			BR	2\$:: GET NEXT CHARACTER
036736	105777	142206		\$TYPEC: TSTB	2\$TPS	:: WAIT UNTIL PRINTER IS READY
036742	100375			BPL	\$TYPEC	
036744	116677	000002	142200	MOVB	2(SP), 2\$TPB	:: LOAD CHAR TO BE TYPED INTO DATA REG.
036752	122766	000015	000002	CMPB	#CR, 2(SP)	:: IS CHARACTER A CARRIAGE RETURN?
036760	001003			BNE	1\$:: BRANCH IF NO
036762	105037	037002		CLRB	\$CHARCNT	:: YES--CLEAR CHARACTER COUNT
036766	000406			BR	\$TYPEX	:: EXIT
036770	122766	000012	000002	1\$: CMPB	#LF, 2(SP)	:: IS CHARACTER A LINE FEED?
036776	001402			BEQ	\$TYPEX	:: BRANCH IF YES
037000	105227			INCB	(PC)+	:: COUNT THE CHARACTER
037002	000000			\$CHARCNT: .WORD	0	:: CHARACTER COUNT STORAGE
037004	000207			\$TYPEX: RTS	PC	

```

9997
9998
9999
0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031

```

```

.SBTTL TTY INPUT ROUTINE

:*****
.ENABL LSB
$TKCNT: .WORD 0          ;;NUMBER OF ITEMS IN QUEUE
$TKQIN: .WORD 0          ;;INPUT POINTER
$TKQOUT: .WORD 0         ;;OUTPUT POINTER
$TKQSRT: .BLKB 9.        ;;TTY KEYBOARD QUEUE
$TKQEND=.
.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

$TKINT: CLR $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
        MOV $TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
        MOV $TKQIN,$TKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
        MOV $TKSRV,$TKVEC  ;;INITIALIZE THE KEYBOARD VECTOR
        MOV #200,$TKVEC+2  ;;"BR" LEVEL 4
        TST $TKB           ;;CLEAR DONE FLAG
        MOV #100,$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
        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)

$TKSRV: MOVB $TKB,-(SP)    ;;PICKUP THE CHARACTER
        BIC #↑C177,(SP)   ;;STRIP THE JUNK
        CMP (SP),#3       ;;IS IT A CONTROL C?
        BNE 1$           ;;BRANCH IF NO
        TYPE $CNTLC       ;;TYPE A CONTROL-C (↑C)
        JSR PC,$TKINT     ;;INIT THE KEYBOARD
        TST (SP)+         ;;CLEAN UP STACK
        JMP OPERSEL       ;;CONTROL C RESTART
1$: CMP (SP),#7           ;;IS IT A CONTROL G?
   BNE 2$               ;;BRANCH IF NO
   CMP #SWREG,SWR       ;;IS SOFT-SWR SELECTED?
   BEQ 6$               ;;GO TO SWR CHANGE
2$: CMP #9,$TKCNT        ;;IS THE QUEUE FULL?
   BNE 3$               ;;BRANCH IF NO
   TYPE $BELL           ;;RING THE TTY BELL
   TST (SP)+           ;;CLEAN CHARACTER OFF OF STACK
   BR 5$                ;;EXIT

```

```

10032 037170 021627 000023 3$: CMP (SP),#23 ;; IS IT A CONTROL-S?
10033 037174 001021 BNE 32$ ;; BRANCH IF NO
10034 037176 005077 141742 CLR $STKS ;; DISABLE TTY KEYBOARD INTERRUPTS
10035 037202 005726 TST (SP)+ ;; CLEAN CHAR OFF STACK
10036 037204 105777 141734 31$: TSTB $STKS ;; WAIT FOR A CHAR
10037 037210 100375 BPL 31$ ;; LOOP UNTIL ITS THERE
10038 037212 117746 141730 MOVB $STKB,-(SP) ;; GET THE CHARACTER
10039 037216 042716 177600 BIC #10177,(SP) ;; MAKE IT 7-BIT ASCII
10040 037222 022627 000021 CMP (SP)+,#21 ;; IS IT A CONTROL-Q?
10041 037226 001366 BNE 31$ ;; BRANCH IF NO
10042 037230 012777 000100 141706 MOV #100,$STKS ;; REENABLE TTY KEYBOARD INTERRUPTS
10043 037236 000002 RTI ;; RETURN
10044 037240 005237 037006 32$: INC $TKCNT ;; COUNT THIS CHARACTER
10045 037244 021627 000140 CMP (SP),#140 ;; IS IT UPPER CASE?
10046 037250 002405 BLT 4$ ;; BRANCH IF YES
10047 037252 021627 000175 CMP (SP),#175 ;; IS IT A SPECIAL CHAR?
10048 037256 003002 BGT 4$ ;; BRANCH IF YES
10049 037260 042716 000040 BIC #40,(SP) ;; MAKE IT UPPER CASE
10050 037264 112677 177520 4$: MOVB (SP)+,$STKQIN ;; AND PUT IT IN QUEUE
10051 037270 005237 037010 INC $TKQIN ;; UPDATE THE POINTER
10052 037274 023727 037010 037025 CMP $TKQIN,$STKQEND ;; GO OFF THE END?
10053 037302 001003 BNE 5$ ;; BRANCH IF NO
10054 037304 012737 037014 037010 5$: MOV # $STKQSR,$STKQIN ;; RESET THE POINTER
10055 037312 000002 RTI ;; RETURN

```

```

10056
10057 *****
10058 *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
10059 *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
10060 *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
10061 *CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

10062 037314 022737 000176 001140 $OKSWR: CMP #SWREG,SWR ;; IS THE SOFT-SWR SELECTED
10063 037322 001124 BNE 15$ ;; EXIT IF NOT
10064 037324 105777 141614 TSTB $STKS ;; IS A CHAR WAITING?
10065 037330 100121 BPL 15$ ;; IF NOT, EXIT
10066 037332 117746 141610 MOVB $STKB,-(SP) ;; YES
10067 037336 042716 177600 BIC #10177,(SP) ;; MAKE IT 7-BIT ASCII
10068 037342 021627 000007 CMP (SP),#7 ;; IS IT A CONTROL-G?
10069 037346 001300 BNE 2$ ;; IF NOT, PUT IT IN THE TTY QUEUE
10070 AND EXIT

```

```

10071 *****
10072 *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
10073 *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
10074 *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

10075 037350 123727 001134 000001 6$: CMPB $AUTOB,#1 ;; ARE WE RUNNING IN AUTO-MODE?
10076 037356 001674 BEQ 2$ ;; BRANCH IF YES
10077 037360 005726 TST (SP)+ ;; CLEAR CONTROL-G OFF STACK
10078 037362 004737 037026 JSR PC,$TKINT ;; FLUSH THE TTY INPUT QUEUE
10079 037366 005077 141552 CLR $STKS ;; DISABLE TTY KEYBOARD INTERRUPTS
10080 037372 112737 000001 001135 MOVB #1,$INTAG ;; SET INTERRUPT MODE INDICATOR
10081
10082
10083 037400 104400 040077 SGTSWR: TYPE , $CNTLG ;; ECHO THE CONTROL-G (IG)
10084 037404 104400 040104 TYPE $MSWR ;; TYPE CURRENT CONTENTS
10085 037410 013746 000176 MOV $SWREG,-(SP) ;; SAVE SWREG FOR TYPEOUT

```

10096	037414	104401			TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
10097	037416	104400	040115		TYPE	.\$MNEW	::PROMPT FOR NEW SWR
10098	037422	005046		19\$:	CLR	-(SP)	::CLEAR COUNTER
10099	037424	005046			CLR	-(SP)	::THE NEW SWR
10090	037426	105777	141512	7\$:	TSTB	2\$TKS	::CHAR THERE?
10091	037432	100375			BPL	7\$::IF NOT TRY AGAIN
10092							
10093	037434	117746	141506		MOVB	2\$TKB, -(SP)	::PICK UP CHAR
10094	037440	042716	177600		BIC	#10177, (SP)	::MAKE IT 7-BIT ASCII
10095							
10096	037444	021627	000003		CMP	(SP), #3	::IS IT A CONTROL-C?
10097	037450	001015			BNE	9\$::BRANCH IF NOT
10098	037452	104400	040065		TYPE	.\$CNTLC	::YES, ECHO CONTROL-C (↑C)
10099	037456	062706	000006		ADD	#6, SP	::CLEAN UP STACK
10100	037462	123727	001135	000001	CMPB	\$INTAG, #1	::REENABLE TTY KEYBOARD INTERRUPTS?
10101	037470	001003			BNE	9\$::BRANCH IF NO
10102	037472	012777	000100	141444	MOV	#100, 2\$TKS	::ALLOW TTY KEYBOARD INTERRUPTS
10103	037500	000137	034140	9\$:	JMP	OPERSEL	::CONTROL-C RESTART
10104							
10105							
10106	037504	021627	000025	9\$:	CMP	(SP), #25	::IS IT A CONTROL-U?
10107	037510	001005			BNE	10\$::BRANCH IF NOT
10108	037512	104400	040072		TYPE	.\$CNTLU	::YES, ECHO CONTROL-U (↑U)
10109	037516	062706	000006	20\$:	ADD	#6, SP	::IGNORE PREVIOUS INPUT
10110	037522	000737			BR	19\$::LET'S TRY IT AGAIN
10111							
10112							
10113	037524	021627	000015	10\$:	CMP	(SP), #15	::IS IT A <CR>?
10114	037530	001022			BNE	15\$::BRANCH IF NO
10115	037532	005766	000004		TST	4(SP)	::YES, IS IT THE FIRST CHAR?
10116	037536	001403			BEQ	11\$::BRANCH IF YES
10117	037540	015677	000002	141372	MOV	2(SP), 2\$SWR	::SAVE NEW SWR
10118	037546	062706	000006	11\$:	ADD	#6, SP	::CLEAN UP STACK
10119	037552	104400	001223	14\$:	TYPE	.\$CRLF	::ECHO <CR> AND <LF>
10120	037556	123727	001135	000001	CMPB	\$INTAG, #1	::RE-ENABLE TTY KBD INTERRUPTS?
10121	037564	001003			BNE	15\$::BRANCH IF NOT
10122	037566	012777	000100	141350	MOV	#100, 2\$TKS	::RE-ENABLE TTY KBD INTERRUPTS
10123	037574	000002		15\$:	RTI		::RETURN
10124	037576	004737	036736	16\$:	JSR	PC, \$TYPEC	::ECHO CHAR
10125	037602	021627	000060		CMP	(SP), #60	::CHAR < C?
10126	037606	002420			BLT	18\$::BRANCH IF YES
10127	037610	021627	000067		CMP	(SP), #67	::CHAR > ??
10128	037614	003015			BGT	18\$::BRANCH IF YES
10129	037616	042726	000060		BIC	#50, (SP)+	::STRIP-OFF ASCII
10130	037622	005766	000002		TST	2(SP)	::IS THIS THE FIRST CHAR
10131	037626	001403			BEQ	17\$::BRANCH IF YES
10132	037630	006316			ASL	(SP)	::NO, SHIFT PRESENT
10133	037632	006316			ASL	(SP)	::CHAR OVER TO MAKE
10134	037634	006316			ASL	(SP)	::ROOM FOR NEW ONE.
10135	037636	005266	000002	17\$:	INC	2(SP)	::KEEP COUNT OF CHAR
10136	037642	056616	177776		BIS	-2(SP), (SP)	::SET IN NEW CHAR
10137	037646	000667			BR	7\$::GET THE NEXT ONE
10138	037650	104400	001222	18\$:	TYPE	.\$QUES	::TYPE ?<CR><LF>
10139	037654	000720			BR	20\$::SIMULATE CONTROL-U

```

10140          .DSABL  LSB
10141
10142
10143          ;*****
10144          ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10145          ;*CALL:
10146          ;*      RDCHR          ;;GET A CHARACTER FROM THE QUEUE
10147          ;*      RETURN HERE    ;;CHARACTER IS ON THE STACK
10148          ;*                  ;;WITH PARITY BIT STRIPPED OFF
10149          ;*
10150
10151 037656 011646 $RDCHR: MOV      (SP),-(SP)    ;;PUSH DOWN THE PC AND
10152 037650 016666 000004 000002      MOV      4(SP),2(SP)    ;;THE PS
10153 037666 005066 000004          CLR      4(SP)        ;;GET READY FOR A CHARACTER
10154 037672 005046          CLR      -(SP)        ;;PUT NEW PS ON STACK
10155 037674 012746 037702          MOV      #64$,-(SP)    ;;PUT NEW PC ON STACK
10156 037700 000002          RTI          ;;POP NEW PC AND PS
10157 037702
10158 037702 005737 037006 64$:      TST      $TKCNT      ;;WAIT ON A CHARACTER
10159 037706 001775 1$:          BEQ      1$
10160 037710 005337 037006          DEC      $TKCNT      ;;DECREMENT THE COUNTER
10161 037714 117766 177072 000004      MOVVB   2$TKQOUT,4(SP) ;;GET ONE CHARACTER
10162 037722 005237 037012          INC      $TKQOUT    ;;UPDATE THE POINTER
10163 037726 023727 037012 037025      CMP     $TKQOUT,#$TKGEND ;;DID IT GO OFF OF THE END?
10164 037734 001003          BNE     2$          ;;BRANCH IF NO
10165 037736 012737 037014 037012      MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10166 037744 000002          RTI          ;;RETURN
10167          ;*****
10168          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10169          ;*CALL:
10170          ;*      RDLIN          ;;INPUT A STRING FROM THE TTY
10171          ;*      RETURN HERE    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10172          ;*                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10173          ;*
10174 037746 010346 $RDLIN: MOV      R3, -(SP)    ;;SAVE R3
10175 037750 012703 040054 1$:      MOV     #$TTYIN,R3    ;;GET ADDRESS
10176 037754 022703 040065 2$:      CMP     #$TTYIN+9.,R3  ;;BUFFER FULL?
10177 037760 101405          BLOS   4$          ;;BR IF YES
10178 037762 104407          RDCHR   ;;GO READ ONE CHARACTER FROM THE TTY
10179 037764 112613          MOVVB  (SP)+,(R3)    ;;GET CHARACTER
10180 037766 122713 000177 10$:    CMPB   #177,(R3)    ;;IS IT A RUBOUT
10181 037772 001003          BNE   3$          ;;SKIP IF NOT
10182 037774 104400 001222 4$:      TYPE   $QUES      ;;TYPE A .?
10183 040000 000763          BR     1$          ;;CLEAR THE BUFFER AND LOOP
10184 040002 111337 040052 3$:      MOVVB  (R3),9$     ;;ECHO THE CHARACTER
10185 040006 104400 040052          TYPE   9$
10186 040012 122723 000015          CMPB   #15,(R3)+   ;;CHECK FOR RETURN
10187 040016 001356          BNE   2$          ;;LOOP IF NOT RETURN
10188 040020 105063 177777          CLRB  -1(R3)      ;;CLEAR RETURN (THE 15)
10189 040024 104400 001224          TYPE   $LF        ;;TYPE A LINE FEED
10190 040030 012603          MOV   (SP)+,R3    ;;RESTORE R3
10191 040032 011646          MOV   (SP),-(SP)  ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10192 040034 016666 000004 000002      MOV   4(SP),2(SP) ;;FIRST ASCII CHARACTER ON IT
10193 040042 012766 040054 000004      MOV   #$TTYIN,4(SP)

```

```

10194 040050 000002          R/I          ;;RETURN
10195 040052      000          9$: .BYTE 0      ;;STORAGE FOR ASCII CHAR. TO TYPE
10196 040053      000          .BYTE 0      ;;TERMINATOR
10197 040054 000011          $TTYIN: .BLKB 9.  ;;RESERVE 9. BYTES FOR TTY INPJT
10198 040065      136 006503 000012 $CNTLC: .ASCIZ /C<<15><12>  ;;CONTROL "C"
10199 040072 052536 005015      000 $CNTLU: .ASCIZ /U<<15><12>  ;;CONTROL "U"
10200 040077      136 006507 000012 $CNTLG: .ASCIZ /G<<15><12>  ;;CONTROL "G"
10201 040104 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
10202 040112 020075      000
10203 040115      040 047040 053505 $MNEW: .ASCIZ / NEW = /
10204 040122 036440 000040
10205

```

;FROM THE TTY


```

10206
10207
10208
10209
10210
10211
10212
10213
10214
10215
10216
10217
10218
10219
10220
10221
10222
10223
10224
10225
10226
10227
10228
10229
10230
10231
10232
10233
10234
10235
10236
10237
10238
10239
10240
10241
10242
10243
10244
10245
10246
10247
10248
10249
10250
10251
10252
10253
10254
10255
10256
10257
10258
10259
10260
10261
10262
10263
10264

```

```

.SBTTL READ AN OCTAL NUMBER FROM THE TTY

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

SRDOCT: MOV      (SP), -(SP)      ;; PROVIDE SPACE FOR THE
        MOV      4(SP), 2(SP)    ;; INPUT NUMBER
        MOV      R0, -(SP)       ;; PUSH R0 ON STACK
        MOV      R1, -(SP)       ;; PUSH R1 ON STACK
        MOV      R2, -(SP)       ;; PUSH R2 ON STACK
15:     RDLIN      ;; READ AN ASCII LINE
        MOV      (SP)+, R0        ;; GET ADDRESS OF 1ST CHARACTER
        MOV      R0, 55         ;; AND SAVE IT
        CLR      R1             ;; CLEAR DATA WORD
        CLR      R2
25:     MOV      (R0)+, -(SP)     ;; PICKUP THIS CHARACTER
        BEQ      35             ;; IF ZERO GET OUT
        CMP      #'0, (SP)      ;; MAKE SURE THIS CHARACTER
        BGT      45             ;; IS AN OCTAL DIGIT
        CMP      #'7, (SP)
        BLT      45
        ASL      R1             ;; *2
        ROL      R2
        ASL      R1             ;; *4
        ROL      R2
        ASL      R1             ;; *8
        ROL      R2
        BIC      #'07, (SP)     ;; STRIP THE ASCII JUNK
        ADD      (SP)+, R1      ;; ADD IN THIS DIGIT
        BR      25             ;; LOOP
35:     TST      (SP)+          ;; CLEAN TERMINATOR FROM STACK
        MOV      R1, 12(SP)     ;; SAVE THE RESULT
        MOV      R2, $HIOCT
        MOV      (SP)+, R2      ;; POP STACK INTO R2
        MOV      (SP)+, R1      ;; POP STACK INTO R1
        MOV      (SP)+, R0      ;; POP STACK INTO R0
        RTI                    ;; RETURN
45:     TST      (SP)+          ;; CLEAN PARTIAL FROM STACK
        CLAB      (R0)          ;; SET A TERMINATOR
        TYPE      ;; TYPE UP THRU THE BAD CHAR.
55:     .WORD    0
        TYPE      $QUES        ;; "?" "CR" & "LF"
        BR      15            ;; TRY AGAIN
$HIOCT: .WORD    0            ;; HIGH ORDER BITS GO HERE

```

```

10259          .SBTTL  ERROR HANDLER ROUTINE
10260
10261          ::*****
10262          ::*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
10263          ::*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10264          ::*AND GO TO $ERRTYP ON ERROR
10265          ::*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10266          ::*SW15=1      HALT ON ERROR
10267          ::*SW13=1      INHIBIT ERROR TYPEOUTS
10268          ::*SW10=1      BELL ON ERROR
10269          ::*SW09=1      LOOP ON ERROR
10270          ::*CALL
10271          ::*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
10272
10273          040266          $ERROR:      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
10274          040266          104406
10275
10276          040270          REGSAV:      MOV      #-1,2,$ERFLG      ;SET ERROR FLAG
10277          040270          012737      177777      004632
10278          040276          REGSA1:
10279
10280          040276          105237      001103      7$:      INCB      $ERFLG      ;;SET THE ERROR FLAG
10281          040302          001775          BEQ      7$      ;;DON'T LET THE FLAG GO TO ZERO
10282          040304          013777      001102      140530      MOV      $STNM,2$DISPLAY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
10283          040312          032777      002000      140620      BIT      #BIT10,3$SWR      ;;BELL ON ERROR?
10284          040320          001402          BEQ      1$      ;;NO - SKIP
10285          040322          104400      001216          TYPE      $BELL      ;;RING BELL
10286          040326          005237      001112          1$:      INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
10287          040332          011637      001115          MOV      (SP),3$ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
10288          040336          162737      000002      001116          SUB      #2,3$ERRPC
10289          040344          117737      140546      001114          MOV      2$ERRPC,1$ITEMB      ;;STRIP AND SAVE THE ERROR ITEM CODE
10290          040352          032777      020000      140560          BIT      #BIT13,2$SWR      ;;SKIP TYPEOUT IF SET
10291          040360          001004          BNE      20$      ;;SKIP TYPEOUTS
10292          040362          004737      040436          JSR      PC,3$ERRTYP      ;;GO TO USER ERROR ROUTINE
10293          040366          104400      001223          TYPE      .3$RLF
10294          040372
10295          040372          005777      140542          20$:      TST      2$SWR      ;;HALT ON ERROR
10296          040376          100002          2$:      BPL      3$      ;;SKIP IF CONTINUE
10297          040400          000000          HALT
10298          040402          104406          CKSWR      ;;HALT ON ERROR!
10299          040404          032777      001000      140526          3$:      BIT      #BIT09,2$SWR      ;;TEST FOR CHANGE IN SOFT-SWR
10300          040412          001402          BEQ      4$      ;;LOOP ON ERROR SWITCH SET?
10301          040414          013716      001110          MOV      $LPERR,(SP)      ;;BR IF NO
10302          040420          005737      001214          4$:      TST      $ESCAPE      ;;FUJGE RETURN FOR LOOPING
10303          040424          001402          BEQ      5$      ;;CHECK FOR AN ESCAPE ADDRESS
10304          040426          013716      001214          MOV      $ESCAPE,(SP)      ;;BR IF NONE
10305          040432
10306          040432          000002          5$:      RTI      ;;FUJGE RETURN ADDRESS FOR ESCAPE

```

.....

.SBTT. ERROR MESSAGE TIMEOUT ROUTINE

*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (ITEMB), TO DETERMINE WHICH
 *ERROR IS TO BE REPORTED. IT THEN OBTAINS FROM THE "ERROR TABLE" SEERRB,
 *AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 *IT IS A COPY OF THE SEERTYP 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.

```

04047 000300
04048 013746 140476
04049 040476 177277
04050 022716 000100
04051 001301
04052 000401
04053 000127 041376
04056 000127 041376
040562
040462 104400 040470
040468 000406
040504 013746 002254
040504 104401
040510
040512 104400 040520
040516 000406
040524 013746 002256
040530 104401
040542 104400 040550
040546 000406
040564 013746 002260
040564 104401
040570
040572 104400 040600
040576 000406
  
```

```

PRITEM: 0 ;PREVIOUS ITEM NO. LOCATION
SEERTYP: MOV 2SWR, -(SP) ;GET SWITCH SETTING
          BIC 2YC5C0, (SP) ;KEEP ONLY SWITCH 8 AND 6
          CMP 2SW06, (SP)+ ;IS 6 SET AND 8 RESET
          BNE 18 ;IF NOT BRANCH
          BR 25 ;BRANCH IF SW 6 IS SET AND 8 RESET
18: JMP 28TYPERR ;JUMP IF SW 8 IS SET
25: ;OR IF SW 8 IS RESET AND SW 6 IS RESET

          TYPE 655 ;:TYPE ASCIZ STRING
          BR 645 ;:GET OVER THE ASCIZ
::655: .ASCIZ <15><12>/RHWC = /
645: MOV 2RWC, -(SP) ;GET READY TO TYPE RHWC CONTENTS
      TYPOC

          TYPE 675 ;:TYPE ASCIZ STRING
          BR 665 ;:GET OVER THE ASCIZ
::675: .ASCIZ <15><12>/RHBA = /
665: MOV 2RBA, -(SP) ;GET READY TO TYPE RHBA CONTENTS
      TYPOC

          TYPE 695 ;:TYPE ASCIZ STRING
          BR 685 ;:GET OVER THE ASCIZ
::695: .ASCIZ <15><12>/RHCS2 = /
685: MOV 2RCS2, -(SP) ;GET READY TO TYPE RHCS2 CONTENTS
      TYPOC

          TYPE 715 ;:TYPE ASCIZ STRING
          BR 705 ;:GET OVER THE ASCIZ
::715: .ASCIZ <15><12>/RHCS1 = /
  
```

REF 5 6 FUNCT. CONT. TST-PT 2
ERROR MESSAGE TYPEOUT ROUTINE

```

040614 013746 002262 70S: MOV 20CS1,-(SP) ;GET READY TO TYPE RHCS1 CONTENTS
040614 104401 TYPOC

040630 104400 040630 TYPE 73S ;:TYPE ASCIZ STRING
040630 000406 BR 72S ;:GET OVER THE ASCIZ
::73S: .ASCIZ <15><12>/RHDS1 = /
72S:

040634 013746 002264 72S: MOV 20DS1,-(SP) ;GET READY TO TYPE RHDS1 CONTENTS
040634 104401 TYPOC

040650 104400 040650 TYPE 75S ;:TYPE ASCIZ STRING
040650 000406 BR 74S ;:GET OVER THE ASCIZ
::75S: .ASCIZ <15><12>/RHER1 = /
74S:

040654 013746 002264 74S: MOV 20ER1,-(SP) ;GET READY TO TYPE RHER1 CONTENTS
040654 104401 TYPOC

040710 104400 040710 TYPE 77S ;:TYPE ASCIZ STRING
040710 000406 BR 76S ;:GET OVER THE ASCIZ
::77S: .ASCIZ <15><12>/RHER2 = /
76S:

040730 013746 002270 76S: MOV 20ER2,-(SP) ;GET READY TO TYPE RHER2 CONTENTS
040730 104401 TYPOC

040740 104400 040740 TYPE 79S ;:TYPE ASCIZ STRING
040736 000406 BR 78S ;:GET OVER THE ASCIZ
::79S: .ASCIZ <15><12>/RHER3 = /
78S:

040754 013746 002276 78S: MOV 20ER3,-(SP) ;GET READY TO TYPE RHER3 CONTENTS
040754 104401 TYPOC

040770 104400 040770 TYPE 81S ;:TYPE ASCIZ STRING
040766 000406 BR 80S ;:GET OVER THE ASCIZ
::81S: .ASCIZ <15><12>/RHDS1 = /
80S:

041004 013746 002266 80S: MOV 20DST,-(SP) ;GET READY TO TYPE RHDS1 CONTENTS
041010 104401 TYPOC

041020 104400 041020 TYPE 83S ;:TYPE ASCIZ STRING
041016 000406 BR 82S ;:GET OVER THE ASCIZ
::83S: .ASCIZ <15><12>/RHCA = /
82S:

041034 013746 002274 82S: MOV 20CA,-(SP) ;GET READY TO TYPE RHCA CONTENTS
041040 104401 TYPOC

041050 104400 041050 TYPE ,85S ;:TYPE ASCIZ STRING

```

041046	000406		BR	84\$::GET OVER THE ASCIZ
041064			BR	84\$::GET OVER THE ASCIZ
041064	013746	002300	MOV	2#AS,-(SP)	:GET READY TO TYPE RHAS CONTENTS
041070	104401		TYPOC		
041072	104400	041100	TYPE	87\$::TYPE ASCIZ STRING
041076	000406		BR	86\$::GET OVER THE ASCIZ
041114			BR	86\$::GET OVER THE ASCIZ
041114	013746	002272	MOV	2#OF,-(SP)	:GET READY TO TYPE RHOF CONTENTS
041122	104400	041130	TYPE	89\$::TYPE ASCIZ STRING
041126	000406		BR	88\$::GET OVER THE ASCIZ
041144			BR	88\$::GET OVER THE ASCIZ
041144	013746	002302	MOV	2#MR,-(SP)	:GET READY TO TYPE RHMR CONTENTS
041150	104401		TYPOC		
041152	104400	041160	TYPE	91\$::TYPE ASCIZ STRING
041156	000406		BR	90\$::GET OVER THE ASCIZ
041174			BR	90\$::GET OVER THE ASCIZ
041174	013746	002320	MOV	2#LA,-(SP)	:GET READY TO TYPE RHLA CONTENTS
041200	104401		TYPOC		
041202	104400	041210	TYPE	93\$::TYPE ASCIZ STRING
041206	000406		BR	92\$::GET OVER THE ASCIZ
041224			BR	92\$::GET OVER THE ASCIZ
041224	013746	002316	MOV	2#CC,-(SP)	:GET READY TO TYPE RHCC CONTENTS
041230	104401		TYPOC		
041232	104400	041240	TYPE	95\$::TYPE ASCIZ STRING
041236	000406		BR	94\$::GET OVER THE ASCIZ
041254			BR	94\$::GET OVER THE ASCIZ
041254	013746	002312	MOV	2#EC1,-(SP)	:GET READY TO TYPE RHEC1 CONTENTS
041260	104401		TYPOC		
041262	104400	041270	TYPE	97\$::TYPE ASCIZ STRING
041266	000406		BR	96\$::GET OVER THE ASCIZ
041304			BR	96\$::GET OVER THE ASCIZ
041304	013746	002314	MOV	2#EC2,-(SP)	:GET READY TO TYPE RHEC2 CONTENTS
041310	104401		TYPOC		

```

041312 104400 041320 TYPE 99$ ::TYPE ASCIZ STRING
041316 000406 BR 98$ ::GET OVER THE ASCIZ
::99$: .ASCIZ <15><12>/RHDT = /
98$: MOV 2#DT,-(SP) ;GET READY TO TYPE RHDT CONTENTS
041334 013746 002306 TYPOC
041334 104401
041342 104400 041350 TYPE 101$ ::TYPE ASCIZ STRING
041346 000406 BR 100$ ::GET OVER THE ASCIZ
::101$: .ASCIZ <15><12>/RMSN = /
100$: MOV 2#SN,-(SP) ;GET READY TO TYPE RMSN CONTENTS
041354 013746 002310 TYPOC
041354 104401
041372 005037 040434 CLR 2#PRITEM ;CLEAR PREVIOUS ERROR ITEM
TYFERR:
041376 104400 001223 TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
041402 010046 MOV R0,-(SP) ;SAVE R0
041404 005000 CLR R0 ;PICKUP THE ITEM INDEX
041406 153700 001114 BISB 2#SITEMB,R0
041412 001004 BNE 1$ ;IF ITEM NUMBER IS ZERO, JUST
;TYPE THE PC OF THE ERROR
041414 013746 001116 MOV $ERRPC,-(SP) ;SAVE $ERRPC FOR TYPEOUT
;ERROR ADDRESS
;GO TYPE--OCTAL ASCII(ALL DIGITS)
;GET OUT
;ADJUST THE INDEX SO THAT IT WILL
;WORK FOR THE ERROR TABLE
1$: TYPOC
BR 10$
2$: DEC R0
ASL R0
ASL R0
ASL R0
ASL R0
ADD #SERRTB,R0 ;FORM TABLE POINTER
CMP R0,2#PRITEM ;WAS PREVIOUS ERROR SAME
BNE 13$ ;BRANCH IF NOT
CMP (R0)+,(R0)+ ;POP R0 OVER EM AND DH
BR 5$
13$: MOV R0,2#PRITEM ;SAVE NEW ERROR ITEM
MOV (R0)+,2$ ;PICKUP "ERROR MESSAGE" POINTER
BEQ 3$ ;SKIP TYPEOUT IF NO POINTER
TYPE ;TYPE THE "ERROR MESSAGE"
WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
3$: TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
MOV (R0)+,4$ ;PICKUP "DATA HEADER" POINTER
BEQ 5$ ;SKIP TYPEOUT IF 0
TYPE ;TYPE THE "DATA HEADER"
WORD 0 ;"DATA HEADER" POINTER GOES HERE
4$: TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
MOV R1,-(SP) ;SAVE R1
MOV (R0)+,R1 ;PICKUP "DATA TABLE" POINTER
BEQ 9$ ;BR IF NO DATA TO BE TYPED
MOV (R0)+,R0 ;PICKUP "DATA FORMAT" POINTER
5$: TSTB (R0)+ ;"OCTAL" OR "DECIMAL"

```

10523	041524	001003		BNE	7\$:BR IF DECIMAL
10524	041526	013146		MOV	2(R1)+,-(SP)		:SAVE 2(R1)+ FOR TYPEOUT
10525	041530	104401		TYPOC			:GO TYPE--OCTAL ASCII(ALL DIGITS)
10526	041532	000402		BR	8\$		
10527	041534		7\$:				
10528	041534	013146		MOV	2(R1)+,-(SP)		:SAVE 2(R1)+ FOR TYPEOUT
10529	041536	104404		TYPDS			:GO TYPE--DECIMAL ASCII WITH SIGN
10530	041540	005711	8\$:	TST	(R1)		:IS THERE ANOTHER NUMBER?
10531	041542	001403		GEQ	9\$:BR IF NO
10532	041544	104400	041560	TYPE	11\$:TYPE TWO(2) SPACES
10533	041550	000764		BR	6\$:LOOP
10534							
10535	041552	012601	9\$:	MOV	(SP)+,R1		:RESTORE R1
10536	041554	012600	10\$:	MOV	(SP)+,R0		: "CARRIAGE RETURN" & "LINE FEED"
10537	041555	000207		RTS	PC		:RETURN
10538	041550	020040	003	11\$:	.ASCIZ / /		:TWO(2) SPACES
10539		041564			.EVEN		

10540
10541
10542
10543
10544
10545
10546
10547
10548
10549
10550
10551
10552
10553
10554
10555
10556
10557
10558
10559
10560
10561
10562
10563
10564
10565
10566
10567
10568
10569
10570
10571
10572
10573
10574
10575
10576
10577
10578
10579
10580
10581
10582
10583
10584
10585
10586
10587
10588
10589
10590
10591
10592
10593

.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.
*STYPOS---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

*STYPOJ---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR STYPOC
*CALL:
* MOV NUM,-(SP) ::NUMBER TO BE TYPED
* TYPOJ ::CALL FOR TYPEOUT

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

041564 017646 000000
041570 116637 000001 042007
041576 112637 042011
041602 062716 000002
041606 000406
041610 112737 000001 042007
041616 112737 000006 042011
041624 112737 000005 042006
041632 010346
041634 010446
041636 010546
041640 113704 042011
041644 005404
041646 062704 000006
041652 110437 042010
041656 113704 042007
041662 016605 000012
041666 005003
041670 006105
041672 000404
041674 006105
041676 006105
041700 006105
041702 010503
041704 006103
041706 105337 042010
041712 100016
041714 042703 177770

STYPOS: MOV 2(SP),-(SP) ::PICKUP THE MODE
 MOVVB 1(SP),%ZFILL ::LOAD ZERO FILL SWITCH
 MOVVB (SP)+,%MODE+1 ::NUMBER OF DIGITS TO TYPE
 ADD #2,(SP) ::ADJUST RETURN ADDRESS
 BR \$TYPOJ
STYPOC: MOVVB #1,%ZFILL ::SET THE ZERO FILL SWITCH
 MOVVB #6,%MODE+1 ::SET FOR SIX(6) DIGITS
STYPOJ: MOVVB #5,%OCNT ::SET THE ITERATION COUNT
 MOV R3,-(SP) ::SAVE R3
 MOV R4,-(SP) ::SAVE R4
 MOV R5,-(SP) ::SAVE R5
 MOVVB %MODE+1,R4 ::GET THE NUMBER OF DIGITS TO TYPE
 NEG R4
 ADD #6,R4 ::SUBTRACT IT FOR MAX. ALLOWED
 MOVVB R4,%MODE ::SAVE IT FOR USE
 MOVVB %ZFILL,R4 ::GET THE ZERO FILL SWITCH
 MOV 12(SP),R5 ::PICKUP THE INPUT NUMBER
 CLR R3 ::CLEAR THE OUTPUT WORD
1\$: ROL R5 ::ROTATE MSB INTO "C"
 BR 2\$::GO DO MSB
2\$: ROL R5 ::FORM THIS DIGIT
 ROL R5
 ROL R5
 MOV R5,R3
3\$: ROL R3 ::GET LSB OF THIS DIGIT
 DECB %MODE ::TYPE THIS DIGIT?
 BPL 7\$::BR IF NO
 BIC #177770,R3 ::GET RID OF JUNK

10594	041720	001002		BNE	4\$::TEST FOR 0
10595	041722	005704		TST	R4	::SUPPRESS THIS 0?
10596	041724	001403		BEQ	5\$::BR IF YES
10597	041726	005204	4\$:	INC	R4	::DON'T SUPPRESS ANYMORE 0'S
10598	041730	052723		BIS	#'0,R3	::MAKE THIS DIGIT ASCII
10599	041734	052703	5\$:	BIS	#'R3	::MAKE ASCII IF NOT ALREADY
10600	041740	110337		MJVB	R3,8\$::SAVE FOR TYPING
10601	041744	104400		TYPE	8\$::GO TYPE THIS DIGIT
10602	041750	105337	7\$:	DECB	\$OCNT	::COUNT BY 1
10603	041754	003347		BGT	2\$::BR IF MORE TO DO
10604	041756	002402		BLT	6\$::BR IF DONE
10605	041758	005204		INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
10606	041762	000744		BR	2\$::GO DO THE LAST DIGIT
10607	041764	012605	6\$:	MOV	(SP)+,R5	::RESTORE R5
10608	041766	012604		MOV	(SP)+,R4	::RESTORE R4
10609	041770	012603		MOV	(SP)+,R3	::RESTORE R3
10610	041772	016666	000002 000004	MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
10611	042000	012616		MOV	(SP)+,(SP)	
10612	042002	000002		RTI		::RETURN
10613	042004	000	8\$:	.BYTE	0	::STORAGE FOR ASCII DIGIT
10614	042005	000		.BYTE	00	::TERMINATOR FOR TYPE ROUTINE
10615	042006	000		.BYTE	00	::OCTAL DIGIT COUNTER
10616	042007	000	\$OCNT:	.BYTE	00	::ZERO FILL SWITCH
10617	042010	000000	\$OFILL:	.BYTE	00	::NUMBER OF DIGITS TO TYPE
			\$OMODE:	.WORD	0	

```

10618
10619
10620
10621
10622
10623
10624
10625
10626 042012 010046
10627 042014 016500 000002
10628 042020 005740
10629 042022 111000
10630 042024 006300
10631 042025 016000 042034
10632 042032 000200
10633
10634
10635
10636
10637
10638
10639
10640
10641 042034
10642 042034 036566
10643 042036 041510
10644 042040 041564
10645 042042 041524
10646 042044 036342
10647
10648 042046 037404
10649
10650 042050 037314
10651 042052 037656
10652 042054 037746
10653 042056 040126
10654 042060 033506

```

.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.

```

```

$TRAP:  MOV      RO, -(SP)      ;;SAVE RO
        MOV      2(SP),RO     ;;GET TRAP ADDRESS
        TST      -(RO)        ;;BACKUP BY 2
        MOVB     (RO),RO      ;;GET RIGHT BYTE OF TRAP
        ASL      RO           ;;POSITION FOR INDEXING
        MOV      $TRPAD(RO),RO ;;INDEX TO TABLE
        RTS      RO           ;;GO TO ROUTINE

```

.SBTTL TRAP TABLE

```

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

```

ROUTINE

```

:-----
$TRPAD: $TYPE   ;;CALL=TYPE   TRAP+0(104400)  TTY TYPEOUT ROUTINE
        $TYPOC  ;;CALL=TYPOC TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS  ;;CALL=TYPOS TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON  ;;CALL=TYPON TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS  ;;CALL=TYPDS TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)

        $GTSWR  ;;CALL=GTSWR TRAP+5(104405)  GET SOFT-SWR SETTING

        $CKSWR  ;;CALL=CKSWR TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
        $RDCHR  ;;CALL=RDCHR TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
        $RDLIN  ;;CALL=RDLIN TRAP+10(104410) TTY TYPEIN STRING ROUTINE
        $RDOCT  ;;CALL=RDOCT TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
        WAIT.T  ;;CALL=WAT   TRAP+12(104412) DONT ADD ABOVE THIS TRAP

```

.SBTTL POWER DOWN AND UP ROUTINES

```

10655
10656
10657
10658
10659 0422062 012737 042226 000024
10660 0422070 012737 000340 000026
10661 0422076 010046
10662 0422100 010146
10663 0422102 010246
10664 0422104 010346
10665 0422106 010446
10666 0422110 010546
10667 0422112 017746 137022
10668 0422116 010637 042232
10669 0422122 012737 042134 000024
10670 0422130 000000
10671 0422132 000776
10672
10673
10674
10675 042134 012737 042226 000024
10676 042142 013706 042232
10677 042146 005037 042232
10678 042152 005237 042232
10679 042156 001375
10680 042160 012677 136754
10681 042164 012605
10682 042166 012604
10683 042170 012603
10684 042172 012602
10685 042174 012601
10686 042176 012600
10687 042200 012737 042062 000024
10688 042206 012737 000340 000026
10689 042214 104400
10690 042216 042234
10691 042220 012716
10692 042222 004710
10693 042224 000002
10694 042226 000000
10695 042230 000776
10696 042232 000000
10697 042234 005015 047520 042527
10698 042242 000122
10699
10700

```

```

*****
: POWER DOWN ROUTINE

```

```

$PWRDN: MOV $SILLUP, @PWRVEC ;; SET FOR FAST UP
MOV #340, @PWRVEC+2 ;; PRIO:7
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 R4, -(SP) ;; PUSH R4 ON STACK
MOV R5, -(SP) ;; PUSH R5 ON STACK
MOV @JSWR, -(SP) ;; PUSH @JSWR ON STACK
MOV SP, $SAVR6 ;; SAVE SP
MOV @PWRUP, @PWRVEC ;; SET UP VECTOR
HALT
BR .-2 ;; HANG UP

```

```

*****
: POWER UP ROUTINE

```

```

$PWRUP: MOV $SILLUP, @PWRVEC ;; SET FOR FAST DOWN
MOV $SAVR6, SP ;; GET SP
CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
IS: INC $SAVR6 ;; WAIT FOR THE INC
BNE IS ;; OF WORD
MOV (SP)+, @JSWR ;; POP STACK INTO @JSWR
MOV (SP)+, R5 ;; POP STACK INTO R5
MOV (SP)+, R4 ;; POP STACK INTO R4
MOV (SP)+, R3 ;; POP STACK INTO R3
MOV (SP)+, R2 ;; POP STACK INTO R2
MOV (SP)+, R1 ;; POP STACK INTO R1
MOV (SP)+, R0 ;; POP STACK INTO R0
MOV @PWRDN, @PWRVEC ;; SET UP THE POWER DOWN VECTOR
MOV #340, @PWRVEC+2 ;; PRIO:7
TYPE $POWER ;; REPORT THE POWER FAILURE
$PWRMG: .WORD $POWER ;; POWER FAIL MESSAGE POINTER
MOV (PC)+, (SP) ;; RESTART AT BEGIN
$PWRAD: .WORD BEGIN ;; RESTART ADDRESS
RTI
$SILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED
BR .-2 ;; BEFORE THE POWER DOWN WAS COMPLETE
$SAVR6: 0 ;; PUT THE SP HERE
$POWER: .ASCIZ <15><12>"POWER"
.EVEN

```

```

10701
10702 ;*****
10703 ;
10704 ;ERROR AND MESSAGE TABLE CONDIMENTS
10705 ;
10706 ;*****
10707
10708
10709
10710
10711 042244 050122 032060 042040 EM1: .ASCIZ /RPO4 DID NOT INTERRUPT/
10712 042252 042111 047040 052117
10713 042260 044440 052116 051105
10714 042266 052522 052120 000
10715 042273 111 052116 051105 EM2: .ASCIZ /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
10716 042300 052522 052120 042440
10717 042306 040516 046102 020105
10718 042314 044502 020124 047504
10719 042322 047127 041040 052125
10720 042330 042440 050130 041505
10721 042336 042524 020104 044502
10722 042344 020124 044504 020104
10723 042352 047516 020124 042523
10724 042360 000124
10725 042362 050122 032060 042040 EM3: .ASCIZ /RPO4 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
10726 042370 042111 047040 052117
10727 042376 044440 052116 051105
10728 042404 052522 052120 053440
10729 042412 042510 020116 054105
10730 042420 042520 052103 042105
10731 042426 041040 052111 042040
10732 042434 042111 051440 052105
10733 042442 000
10734 042443 105 050130 041505 EM4: .ASCIZ /EXPECTED BIT DID SET BUT TIME IS IN ERROR - TIME IN 10 MICROSEC. DECIMA
10735 042450 042524 020104 044502
10736 042456 020124 044504 020104
10737 042464 042523 020124 052502
10738 042472 020124 044524 042515
10739 042500 044440 020123 047111
10740 042506 042440 051122 051117
10741 042514 026440 052040 046511
10742 042522 020105 047111 030440
10743 042530 020060 044515 051103
10744 042536 051517 041505 020056
10745 042544 042504 044503 040515
10746 042552 000114
10747 042554 044122 051501 042040 EM5: .ASCIZ /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
10748 042562 042517 020123 047516
10749 042570 020124 046103 040505
10750 042576 020122 054502 046440
10751 042604 053117 047111 020107
10752 042612 047111 040440 046114
10753 042620 047440 042516 000123
10754 042626 047514 042101 047111 EM6: .ASCIZ /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/

```

10755	042634	020107	044122	051105
10756	042642	020061	047506	020122
10757	042650	046101	020114	047125
10758	042656	052111	020123	044504
10759	042664	020104	047516	020124
10760	042672	042523	020124	047101
10761	042700	020131	044122	051501
10762	042706	041040	052111	000123
10763	042714	050123	041505	043111
10764	042722	042511	020104	042522
10765	042730	044507	052123	051105
10766	042736	047040	047117	042440
10767	042744	044530	052123	047101
10768	042752	026124	051440	020117
10769	042760	041101	051117	020124
10770	042766	051120	043517	040522
10771	042774	000115		
10772	042776	052123	050117	042520
10773	043004	020104	051104	053111
10774	043012	020105	040510	020123
10775	043020	047515	020114	044502
10776	043026	020124	047111	051040
10777	043034	042110	030523	051440
10778	043042	052105	000	
10779				
10780	043045	127	052111	020110
10781	043052	050123	047111	046104
10782	043060	020105	047520	042527
10783	043066	042522	020104	047504
10784	043074	047127	051040	041510
10785	043102	031123	051440	047510
10786	043110	046125	020104	047117
10787	043116	054514	044040	053101
10788	043124	020105	047125	052111
10789	043132	047040	035117	040440
10790	043140	042116	044440	020122
10791	043146	042523	000124	
10792	043152	043101	042524	020122
10793	043160	050123	047111	046104
10794	043166	020105	047520	042527
10795	043174	042522	020104	050125
10796	043202	020054	047516	050040
10797	043210	041501	020113	041501
10798	043216	047113	020056	044122
10799	043224	051504	020061	044123
10800	043232	052517	042114	044040
10801	043240	053101	020105	047515
10802	043246	036514	026061	053040
10803	043254	036526	000060	
10804	043260	044527	044124	051440
10805	043266	044520	042116	042514
10806	043274	050040	053517	051105
10807	043302	042105	020054	047516
10808	043310	044440	052116	040511

EM7: .ASCIZ /SPECIFIED REGISTER NON EXISTANT. SO ABORT PROGRAM/

EM10: .ASCIZ /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/

EM11: .ASCIZ /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO: AND IR SET/

EM12: .ASCIZ /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1. VV=0/

EM13: .ASCIZ /WITH SPINDLE POWERED, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0. DVA=1, RDY=

```

108009 043316 041514 042532 020054
108010 043324 044122 051503 020061
108011 043332 044123 052517 042114
108012 043340 044040 053101 020105
108013 043346 047507 020075 020054
108014 043354 053104 035501 026061
108015 043362 051040 054504 030475
108016 043370 020054 042511 030075
108017 043376 000
108018 043384 051107 052106 051105 EM14: .ASCIZ /AFTER SPINDLE POWERED UP RHCC SHOULD BE=0/
108019 043392 051440 044520 042116
108020 043400 042514 050040 053517
108021 043408 051105 042105 052440
108022 043416 020120 044122 041503
108023 043424 051440 047510 046125
108024 043432 020104 042502 030075
108025 043440 000
108026 043448 041501 020113 EM15: .ASCII /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR<<15><12>
108027 043456 041501 047113 053517
108028 043464 042514 043504 020105
108029 043472 047503 046515 047101
108030 043480 020104 040503 051525
108031 043488 042105 040440 020116
108032 043496 051105 047522 006522
108033 043504 012
108034 043512 047517 020104 .ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
108035 043520 040504 040524 044440
108036 043528 020123 042502 047506
108037 043536 042522 041440 046517
108038 043544 040515 042116 020054
108039 043552 042522 020103 040504
108040 043560 040524 044440 020123
108041 043568 043101 042524 020122
108042 043576 047503 046515 047101
108043 043584 000104
108044 043592 047516 047455 020120 EM16: .ASCII /NO-OP COMMAND CAUSED AN ERROR<<15><12>
108045 043600 047503 046515 047101
108046 043608 020104 040503 051525
108047 043616 042105 040440 020116
108048 043624 051105 047522 006522
108049 043632 012
108050 043640 047517 020104 .ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
108051 043648 040504 040524 044440
108052 043656 020123 042502 047506
108053 043664 042522 041440 046517
108054 043672 040515 042116 020054
108055 043680 042522 020103 040504
108056 043688 040524 044440 020123
108057 043696 043101 042524 020122
108058 043704 047503 046515 047101
108059 043712 000104
108060 043720 051104 053111 020105 EM17: .ASCII /DRIVE CLEAR COMMAND CAUSED AN ERROR<<15><12>
108061 043728 046103 040505 020122
108062 043736 047503 046515 047101

```

10863	043762	020104	040503	051525
10864	043770	042105	040440	020116
10865	043776	051105	047522	006522
10866	044004	012		
10867	044005	107	047517	020104
10868	044012	040504	040524	043440
10869	044020	053111	051505	051440
10870	044026	047510	046125	020104
10871	044034	042502	020054	042522
10872	044042	020103	040504	040524
10873	044050	043440	053111	051505
10874	044056	040440	052106	051105
10875	044064	041440	046517	040515
10876	044072	042116	000	
10877	044075	122	040505	026504
10878	044102	047111	041440	046517
10879	044110	040515	042116	041440
10880	044116	052501	042523	020104
10881	044124	047101	042440	051122
10882	044132	051117	005015	
10883	044136	047507	042117	042040
10884	044144	052101	020101	044507
10885	044152	042526	020123	044123
10886	044160	052517	042114	041040
10887	044166	026105	051040	041505
10888	044174	042040	052101	020101
10889	044202	044507	042526	020123
10890	044210	042522	027107	041440
10891	044216	047117	042524	052116
10892	044224	020123	043101	042524
10893	044232	020122	047503	046515
10894	044240	047101	000104	
10895				
10896	044244	044122	051503	020061
10897	044252	047503	052116	047105
10898	044260	051524	042040	051125
10899	044266	047111	020107	047503
10900	044274	046515	047101	020104
10901	044302	040527	020123	047111
10902	044310	042440	051122	051117
10903	044316	000		
10904	044317	122	042110	030523
10905	044324	041440	047117	042524
10906	044332	052116	020123	052504
10907	044340	044522	043516	041440
10908	044346	046517	040515	042116
10909	044354	053440	051501	044440
10910	044362	020116	051105	047522
10911	044370	000122		
10912	044372	047125	047514	042101
10913	044400	041440	046517	040515
10914	044406	042116	041440	052501
10915	044414	042523	020104	047101
10916	044422	042440	051122	051117

.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/

EM20: .ASCII /READ-IN COMMAND CAUSED AN ERROR/<15><12>

.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONTENTS AFTER COMMAND/

EM21: .ASCIZ /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/

EM22: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/

EM23: .ASCII /UNLOAD COMMAND CAUSED AN ERROR/<15><12>

POWER DOWN AND UP ROUTINES

Vertical column of alphanumeric characters and symbols, likely a data stream or log output, containing various alphanumeric characters and control characters.

EM27: .ASCII WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE<<15<<12>

.ASCII GOOD DATA GIVES WHAT SHOULD BE THERE<<15<<12>

.ASCIIZ RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND<

EM30: .ASCIIZ WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER<

EM31: .ASCII READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE<<15<<12>

.ASCII GOOD DATA GIVES WHAT SHOULD BE THERE<<15<<12>

MACY11 27(655) 30-MAR-76 18:59 PAGE 239

POWER DOWN AND UP ROUTINES

051440 047510
020104 042502
042510 042522
042503 053111
042040 052101
044500 042526
044122 052101
051501 052040
042520 040440
051105 041440
040515 042116
044522 042524
040500 042524
040500 040524
041111 047511
020104 054502
040500 020104
051101 051105
042116 042040
020101 040503
051105 051105
000122 042040
042101 040503
020101 044440
042105 044440
042522 044520
051105 041440
043516 006505
047517 020104
040524 043440
051505 053440
020124 044123
042114 041040
041124 051105
041505 044505
020104 040524
040440 053111
050440 040510
040522 020122
051105 020105
042524 020122
045515 047101
042101 042040
020101 047111
051122 041505
052111 020105

.ASCIZ RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/

EM32: .ASCIZ WRITE HEADER DATA FOLLOWED BY REAC HEADER AND DATA CAUSED DATA ERROR/

EM33: .ASCII READ DATA CAUSED IMPROPER REGISTER CHANGE <<15>> <12>

.ASCII GOOD DATA GIVES WHAT SHOULD BE THERE <<15>> <12>

.ASCIZ RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND

EM34: .ASCIZ READ DATA INCORRECT/

EM35: .ASCII WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE <15> <12>

MACY11 27(655) 30-MAR-76 18:59 PAGE 240
POWER DOWN AND UP ROUTINES

```

111079 046333 042054 040524 041440
111080 046334 046511 040515 042116
111081 046335 041440 052501 042523
111082 046336 020104 046511 051123
111083 046337 050111 051105 051040
111084 046338 042505 051511 042524
111085 046339 020123 044103 047101
111086 046340 042505 005015
111087 046341 042111 042040
111088 046342 052010 020101 044507
111089 046343 042526 020123 044127
111090 046344 052101 051440 047510
111091 046345 046125 020104 042502
111092 046346 005015 042510 042522
111093 046347 042522 042503 053111
111094 046348 042040 042040 052101
111095 046349 020101 044507 042526
111096 046350 020123 042522 044507
111097 046351 051105 041440
111098 046352 042524 052116
111099 046353 020123 043101 042524
111100 046354 020122 047503 046515
111101 046355 047101 000104
111102 046356 051127 052111 020105
111103 046357 040504 040524 041440
111104 046358 046517 040515 042116
111105 046359 041440 040510 043516
111106 046360 042105 053440 044522
111107 046361 042524 042040 047522
111108 046362 020115 052502 043106
111109 046363 051105 0000
111110 046364 051105 0000
111111 046365 123 042505 020113
111112 046366 047503 046515 047101
111113 046367 020104 040503 051525
111114 046368 042105 044440 050115
111115 046369 047522 042520 020122
111116 046370 042522 044507 052123
111117 046371 051105 041440 040510
111118 046372 043516 006505 012
111119 046373 107 047517 020104
111120 046374 040504 040524 043440
111121 046375 053111 051505 053440
111122 046376 040510 020124 044123
111123 046377 052517 042114 041040
111124 046378 020105 044124 051105
111125 046379 006505 012
111126 046380 122 041505 044505
111127 046381 042526 020104 040504
111128 046382 040524 043440 053111
111129 046383 051505 051040 043505
111130 046384 051511 042524 020122
111131 046385 047503 052116 047105
111132 046386 051524 040440 052106

```

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<<15><12>

.ASCIZ RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/

EM36: .ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/

EM37: .ASCII /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<<15><12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<<15><12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND.

11133	046714	051105	051440	042505	
11134	046722	020113	047503	046515	
11135	046730	047101	000104		
11136	046734	051127	052111	020105	EM40: .ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
11137	046742	044103	041505	020113	
11138	046750	040503	051525	042105	
11139	046756	044440	050115	047522	
11140	046764	042520	020122	042522	
11141	046772	044507	052123	051105	
11142	047000	041440	040510	043516	
11143	047006	006505	012		
11144	047011	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11145	047016	040504	040524	043440	
11146	047024	053111	051505	053440	
11147	047032	040510	020124	044123	
11148	047040	052517	042114	041040	
11149	047046	020105	044124	051105	
11150	047054	006505	012		
11151	047057	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
11152	047064	042526	020104	040504	
11153	047072	040524	043440	053111	
11154	047100	051505	051040	043505	
11155	047106	051511	042524	020122	
11156	047114	047503	052116	047105	
11157	047122	051524	040440	052106	
11158	047130	051105	041440	046517	
11159	047136	040515	042116	000	
11160					
11161	047143	114	041517	044513	EM41: .ASCII /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
11162	047150	043516	047440	052125	
11163	047156	053440	044522	042524	
11164	047164	041040	020131	051127	
11165	047172	052111	020105	047514	
11166	047200	045503	041040	052125	
11167	047206	047524	020116	040503	
11168	047214	051525	042105	044440	
11169	047222	050115	047522	042520	
11170	047230	020122	042522	044507	
11171	047236	052123	051105	041440	
11172	047244	040510	043516	006505	
11173	047252	012			
11174	047253	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11175	047260	040504	040524	043440	
11176	047266	053111	051505	053440	
11177	047274	040510	020124	044123	
11178	047302	052517	042114	041040	
11179	047310	020105	044124	051105	
11180	047316	006505	012		
11181	047321	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
11182	047326	042526	020104	040504	
11183	047334	040524	043440	053111	
11184	047342	051505	051040	043505	
11185	047350	051511	042524	020122	
11186	047356	047503	052116	047105	

11187	047364	051524	040440	052106
11188	047372	051105	053440	044522
11189	047400	042524	020123	042527
11190	047406	042522	045040	041517
11191	047414	042513	020104	052517
11192	047422	000124		
11193	047424	052101	042524	050115
11194	047432	044524	043516	052040
11195	047440	020117	051127	052111
11196	047446	020105	044527	044124
11197	047454	053440	044522	042524
11198	047462	020123	047514	045503
11199	047470	042105	047440	052125
11200	047476	041440	052501	042523
11201	047504	020104	046511	051120
11202	047512	050117	051105	051040
11203	047520	043505	051511	042524
11204	047526	020122	044103	047101
11205	047534	042507	005015	
11206	047540	047507	042117	042040
11207	047546	052101	020101	044507
11208	047554	042526	020123	044127
11209	047562	052101	051440	047510
11210	047570	046125	020104	042502
11211	047576	052040	042510	042522
11212	047604	005015		
11213	047606	042522	042503	053111
11214	047614	042105	042040	052101
11215	047622	020101	044507	042526
11216	047630	020123	042522	044507
11217	047636	052123	051105	041440
11218	047644	047117	042524	052116
11219	047652	020123	043101	042524
11220	047660	020122	052101	042524
11221	047666	050115	042524	020104
11222	047674	051127	052111	000105
11223	047702	051127	052111	047111
11224	047710	020107	044527	044124
11225	047716	053440	044522	042524
11226	047724	020123	047514	045503
11227	047732	042105	047440	052125
11228	047740	041440	040510	043516
11229	047746	042105	042040	051511
11230	047754	020113	040504	040524
11231	047762	005015		
11232	047764	047507	042117	042040
11233	047772	052101	020101	044507
11234	052220	042526	020123	044127
11235	050006	052101	053440	051501
11236	050014	047440	020116	044504
11237	050022	045523	041040	043105
11238	050030	051117	020105	051127
11239	050036	052111	020105	044527
11240	050044	044124	053440	044522

EM42: .ASCII /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<<15><12>

.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/

EM43: .ASCII /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<<15><12>

.ASCII /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE WITH WRITE LOCKED OUT/<<15

11241	050052	042524	046040	041517	
11242	050060	042513	020104	052517	
11243	050066	006524	012		
11244	050071	127	051501	040440	.ASCII /WAS ATTEMPTED/<<15><12>
11245	050076	052124	046505	052120	
11246	050104	042105	005015		
11247	050110	042522	042503	053111	.ASCII /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<<15><12>
11248	050116	042105	042040	052101	
11249	050124	020101	044507	042526	
11250	050132	020123	044127	052101	
11251	050140	053440	051501	051040	
11252	050146	040505	020104	040502	
11253	050154	045503	040440	052106	
11254	050162	051105	053440	044522	
11255	050170	042524	005015		
11256	050174	044527	044124	053440	.ASCIZ /WITH WRITE LOCKED OUT WAS ATTEMPTED/
11257	050202	044522	042524	046040	
11258	050210	041517	042513	020104	
11259	050216	052517	020124	040527	
11260	050224	020123	052101	042524	
11261	050232	050115	042524	000104	
11262	050240	047105	041101	044514	EM44: .ASCII /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<<1
11263	050246	043516	053440	044522	
11264	050254	042524	020123	054502	
11265	050262	053440	044522	042524	
11266	050270	046040	041517	020113	
11267	050276	052502	052124	047117	
11268	050304	041440	052501	042523	
11269	050312	020104	046511	051120	
11270	050320	050117	051105	051040	
11271	050326	043505	051511	042524	
11272	050334	020122	044103	047101	
11273	050342	042507	005015		
11274	050346	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<<15><12>
11275	050354	052101	020101	044507	
11276	050362	042526	020123	044127	
11277	050370	052101	051440	047510	
11278	050376	046125	020104	042502	
11279	050404	052040	042510	042522	
11280	050412	005015			
11281	050414	042522	042503	053111	.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON.<<15><12>
11282	050422	042105	042040	052101	
11283	050430	020101	044507	042526	
11284	050436	020123	042522	044507	
11285	050444	052123	051105	041440	
11286	050452	047117	042524	052116	
11287	050460	020123	043101	042524	
11288	050466	020122	051127	052111	
11289	050474	020105	047514	045503	
11290	050502	041040	052125	047524	
11291	050510	006516	012		
11292	050513	105	040516	046102	.ASCIZ /ENABLED WRITES/
11293	050520	042105	053440	044522	
11294	050526	042524	000123		

11295	050532	051124	047101	043123	EM45: .ASCII /TRANSFERRING ON LAST BLOCK - CYLINDER 410. - 814., SECTOR 21, /<15><12>
11296	050540	051105	044522	043516	
11297	050546	047440	020116	040514	
11298	050554	052123	041040	047514	
11299	050562	045503	026440	041440	
11300	050570	046131	047111	042504	
11301	050576	020122	030464	027060	
11302	050604	026440	034040	032061	
11303	050612	026056	051440	041505	
11304	050620	047524	020122	030462	
11305	050626	020054	005015		
11306	050632	051124	041501	020113	.ASCII /TRACK 18, CAUSED IMPROPER REGISTER CHANGE/<15><12>
11307	050640	034061	020054	040503	
11308	050646	051525	042105	044440	
11309	050654	050115	047522	042520	
11310	050662	020122	042522	044507	
11311	050670	052123	051105	041440	
11312	050676	040510	043516	006505	
11313	050704	012			
11314	050705	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11315	050712	040504	040524	043440	
11316	050720	053111	051505	053440	
11317	050726	040510	020124	044123	
11318	050734	052517	042114	041040	
11319	050742	020105	044124	051105	
11320	050750	006505	012		
11321	050753	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
11322	050760	042526	020104	040504	
11323	050766	040524	043440	053111	
11324	050774	051505	051040	043505	
11325	051002	051511	042524	020122	
11326	051010	047503	052116	047105	
11327	051016	051524	040440	052106	
11328	051024	051105	052040	040522	
11329	051032	051516	042506	000122	
11330	051040	040504	040524	051040	EM46: .ASCII /DATA READ FROM LAST BLOCK - CYLINDER 410. - 814., SECTOR 21, /<15><12>
11331	051046	040505	020104	051106	
11332	051054	046517	046040	051501	
11333	051062	020124	046102	041517	
11334	051070	020113	020055	054503	
11335	051076	044514	042116	051105	
11336	051104	032040	030061	020056	
11337	051112	020055	030470	027064	
11338	051120	020054	042523	052103	
11339	051126	051117	031040	026061	
11340	051134	005015			
11341	051136	051124	041501	020113	.ASCIZ /TRACK 18, IS IN ERROR/
11342	051144	034061	020054	051511	
11343	051152	044440	020116	051105	
11344	051160	047522	000122		
11345	051164	051124	047101	043123	EM47: .ASCII /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
11346	051172	051105	044522	043516	
11347	051200	042040	052101	020101	
11348	051206	051106	046517	047040	

11349	051214	047117	054105	051511
11350	051222	040524	052116	051440
11351	051230	041505	047524	020122
11352	051236	040503	051525	042105
11353	051244	044440	050115	047522
11354	051252	042520	020122	005015
11355	051260	042522	044507	052123
11356	051266	051105	041440	040510
11357	051274	043516	026105	043440
11358	051302	047517	020104	040504
11359	051310	040524	043440	053111
11360	051316	051505	053440	040510
11361	051324	020124	044123	052517
11362	051332	042114	041040	020105
11363	051340	044124	051105	006505
11364	051346	012		
11365	051347	122	041505	044505
11366	051354	042526	020104	040504
11367	051362	040524	043440	053111
11368	051370	051505	051040	043505
11369	051376	051511	042524	020122
11370	051404	047503	052116	047105
11371	051412	051524	040440	052106
11372	051420	051105	040440	052124
11373	051426	046505	052120	042105
11374	051434	052040	040522	051516
11375	051442	042506	000122	
11376	051446	051124	047101	043123
11377	051454	051105	044522	043516
11378	051462	043040	047522	020115
11379	051470	047516	042516	044530
11380	051476	052123	047101	020124
11381	051504	042523	052103	051117
11382	051512	041440	052501	042523
11383	051520	020104	040504	040524
11384	051526	042440	051122	051117
11385	051534	005015		
11386	051536	047507	042117	042040
11387	051544	052101	020101	044507
11388	051552	042526	020123	044127
11389	051560	052101	051440	047510
11390	051566	046125	020104	042502
11391	051574	052040	042510	042522
11392	051602	005015		
11393	051604	040502	020104	040504
11394	051612	040524	043440	053111
11395	051620	051505	053440	040510
11396	051626	020124	040527	020123
11397	051634	047111	041040	043125
11398	051642	042506	020122	043101
11399	051650	042524	020122	051124
11400	051656	047101	043123	051105
11401	051664	000		
11402				

.ASCII /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15/<12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/

EMSO: .ASCII /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15/<12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15/<12>

.ASCIZ /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/

11403	051665	107	053111	047111	EMS1: .ASCII /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/'15'<12>
11404	051672	020107	046111	042514	
11405	051700	040507	020114	052506	
11406	051706	041516	044524	047117	
11407	051714	041440	052501	042523	
11408	051722	020104	046511	051120	
11409	051730	050117	051105	051040	
11410	051736	043505	051511	042524	
11411	051744	020122	044103	047101	
11412	051752	042507	005015		
11413	051756	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11414	051754	052101	020101	044507	
11415	051772	042526	020123	044127	
11416	052000	052101	051440	047510	
11417	052006	046125	020104	042502	
11418	052014	052040	042510	042522	
11419	052022	005015			
11420	052024	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
11421	052032	042105	042040	052101	
11422	052040	020101	044507	042526	
11423	052046	020123	042522	044507	
11424	052054	052123	051105	041440	
11425	052062	047117	042524	052116	
11426	052070	020123	043101	042524	
11427	052076	020122	046111	042514	
11428	052104	040507	020114	052506	
11429	052112	041516	044524	047117	
11430	052120	044440	020123	044507	
11431	052126	042526	000116		
11432	052132	051127	052111	020105	EMS2: .ASCII /WRITE DATA ON NONEXISTANT SECTOR CAUSED IMPROPER REGISTER CHANGE/<15><1
11433	052140	040504	040524	047440	
11434	052146	020116	047516	042516	
11435	052154	044530	052123	047101	
11436	052162	020124	042523	052103	
11437	052170	051117	041440	052501	
11438	052176	042523	020104	046511	
11439	052204	051120	050117	051105	
11440	052212	051040	043505	051511	
11441	052220	042524	020122	044103	
11442	052226	047101	042507	005015	
11443	052234	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11444	052242	052101	020101	044507	
11445	052250	042526	020123	044127	
11446	052256	052101	051440	047510	
11447	052264	046125	020104	042502	
11448	052272	052040	042510	042522	
11449	052300	005015			
11450	052302	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
11451	052310	042105	042040	052101	
11452	052316	020101	044507	042526	
11453	052324	020123	042522	044507	
11454	052332	052123	051105	041440	
11455	052340	047117	042524	052116	
11456	052346	020123	043101	042524	

11457	052354	020122	052101	042524	
11458	052362	050115	042524	020104	
11459	052370	051127	052111	020105	
11460	052376	040504	042524	000	
11461	052403	122	040505	020104	EMS3: .ASCIZ READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
11462	052410	042510	042101	051105	
11463	052416	040440	042116	042040	
11464	052424	052101	020101	043101	
11465	052432	042524	020122	020101	
11466	052440	042523	051101	044103	
11467	052446	041440	052501	042523	
11468	052454	020104	040504	040524	
11469	052462	042440	051122	051117	
11470	052470	000			
11471	052471	101	052124	046505	EMS4: .ASCII /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
11472	052476	052120	047111	020107	
11473	052504	047503	046515	047101	
11474	052512	020104	044527	044124	
11475	052520	044440	053116	046101	
11476	052526	042111	040440	042104	
11477	052534	042522	051523	041440	
11478	052542	052501	042523	020104	
11479	052550	046511	051120	050117	
11480	052556	051105	051040	043505	
11481	052564	051511	042524	020122	
11482	052572	044103	047101	042507	
11483	052600	005015			
11484	052602	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11485	052610	052101	020101	044507	
11486	052616	042526	020123	044127	
11487	052624	052101	051440	047510	
11488	052632	046125	020104	042502	
11489	052640	052040	042510	042522	
11490	052646	005015			
11491	052650	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
11492	052656	042105	042040	052101	
11493	052664	020101	044507	042526	
11494	052672	020123	042522	044507	
11495	052700	052123	051105	041440	
11496	052706	047117	042524	052116	
11497	052714	020123	043101	042524	
11498	052722	020122	050117	051105	
11499	052730	052101	047511	000116	
11500	052736	051127	052111	047111	EMS5: .ASCII /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
11501	052744	020107	051117	051040	
11502	052752	040505	044504	043516	
11503	052760	053440	052111	020110	
11504	052766	054105	042520	052103	
11505	052774	042105	040440	042104	
11506	053002	042522	051523	047440	
11507	053010	042526	043122	047514	
11508	053016	020127	051105	047522	
11509	053024	006522	012		
11510	053027	103	052501	042523	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>

11511	053034	020104	046511	051120
11512	053042	050117	051105	051040
11513	053050	043505	051511	042524
11514	053056	020122	044103	047101
11515	053064	042507	005015	
11516	053070	047507	042117	042040
11517	053076	052101	020101	044507
11518	053104	042526	020123	044127
11519	053112	052101	051440	047510
11520	053120	046125	020104	042502
11521	053126	052040	042510	042522
11522	053134	005015		
11523	053136	042522	042503	053111
11524	053144	042105	042040	052101
11525	053152	020101	044507	042526
11526	053160	020123	042522	044507
11527	053166	052123	051105	041440
11528	053174	047117	042524	052116
11529	053202	020123	043101	042524
11530	053210	020122	050117	051105
11531	053216	052101	047511	000116
11532	053224	040504	040524	051040
11533	053232	040505	020104	044527
11534	053240	044124	040440	020116
11535	053246	054105	042520	052103
11536	053254	042105	040440	042104
11537	053262	042522	051523	047440
11538	053270	042526	043122	047514
11539	053276	020127	051105	047522
11540	053304	020122	051511	044440
11541	053312	041516	051117	042522
11542	053320	052103	005015	
11543	053324	047527	042122	047040
11544	053332	027117	030440	052040
11545	053340	020117	033062	020060
11546	053346	044123	052517	042114
11547	053354	041040	020105	042522
11548	053362	042101	020054	047527
11549	053370	042122	047040	020117
11550	053376	033062	020061	047524
11551	053404	031040	033066	051440
11552	053412	047510	046125	006504
11553	053420	012		
11554	053421	102	020105	044103
11555	053426	047101	042507	000104
11556	053434	052101	042524	050115
11557	053442	044524	043516	042040
11558	053450	052101	020101	047503
11559	053456	046515	047101	020104
11560	053464	044527	044124	053440
11561	053472	047522	043516	043040
11562	053500	051117	040515	020124
11563	053506	044502	020124	040503
11564	053514	051525	042105	005015

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/

EM56: .ASCII /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>

.ASCII /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>

.ASCIZ /BE CHANGED/

EM57: .ASCII /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>

11565	053522	046511	051120	050117	.ASCII	/IMPROPER REGISTER CHANGE/<15><12>
11566	053530	051105	051040	043505		
11567	053536	051511	042524	020122		
11568	053544	044103	047101	042507		
11569	053552	005015				
11570	053554	047507	042117	042040	.ASCII	/GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11571	053552	052101	020101	044507		
11572	053570	042526	020123	044127		
11573	053576	052101	051440	047510		
11574	053604	046125	020104	042502		
11575	053612	052040	042510	042522		
11576	053620	005015				
11577	053622	042522	042503	053111	.ASCIZ	/RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
11578	053630	042105	042040	052101		
11579	053636	020101	044507	042526		
11580	053644	020123	042522	044507		
11581	053652	052123	051105	041440		
11582	053660	047117	042524	052116		
11583	053666	020123	043101	042524		
11584	053674	020122	052101	042524		
11585	053702	050115	042524	020104		
11586	053710	040504	040524	052040		
11587	053716	040522	051516	042506		
11588	053724	000122				
11589	053726	052101	042524	050115	EM60: .ASCII	/ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
11590	053734	044524	043516	052040		
11591	053742	020117	047515	044504		
11592	053750	054506	051040	043505		
11593	053756	051511	042524	020122		
11594	053764	052504	044522	043516		
11595	053772	040440	020116	050117		
11596	054000	051105	052101	047511		
11597	054006	020116	040503	051525		
11598	054014	042105	044440	050115		
11599	054022	047522	042520	006522		
11600	054030	012				
11601	054031	122	043505	051511	.ASCII	/REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11602	054036	042524	020122	044103		
11603	054044	047101	042507	020056		
11604	054052	047507	042117	042040		
11605	054060	052101	020101	044507		
11606	054066	042526	020123	044127		
11607	054074	052101	051440	047510		
11608	054102	046125	020104	042502		
11609	054110	052040	042510	042522		
11610	054116	005015				
11611	054120	042522	042503	053111	.ASCII	/RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
11612	054126	042105	042040	052101		
11613	054134	020101	044507	042526		
11614	054142	020123	042522	044507		
11615	054150	052123	051105	041440		
11616	054156	047117	042524	052116		
11617	054164	020123	043101	042524		
11618	054172	020122	050117	051105		

.ASCII MAPPING REG GIVES ADDRESS OF REGISTER BEING MODIFIED AND CALLED FROM

EM61: .ASCII DEVICE NOT AVAILABLE BEFORE COMMAND WAS TO BE GIVEN

EM63: .ASCII RHDS1 CONTENTS DURING COMMAND WAS IN ERROR.

EM64: .ASCII RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE (15) (12)

.ASCII GOOD DATA GIVES WHAT SHOULD BE THERE (15) (12)

.ASCII RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND

Vertical column of binary and hexadecimal data on the left side of the page, including labels like 'EM61', 'EM63', and 'EM64'.

MINI-11-22RJJ-A RFO4 5 6 FUNCT. CONT. TST-PT 2
POWER DOWN AND LP ROUTINES

11701	055316	031040	030066	040440
11702	055324	042522	042040	052101
11703	055332	020101	047527	042122
11704	055340	005123	015	
11705	055348	111	020116	040504
11706	055350	040524	053440	051117
11707	055356	051504	041040	052111
11708	055364	020123	026064	026065
11709	055372	026066	026067	020070
11710	055400	044507	042526	052040
11711	055406	040522	045503	047040
11712	055414	046525	042502	000122
11713	055422	042522	042101	044040
11714	055430	040505	042504	020122
11715	055436	047101	020104	040504
11716	055444	040524	042440	051122
11717	055452	051117	044440	006516
11718	055460	012		
11719	055461	104	043111	042506
11720	055466	042522	041516	020105
11721	055474	044514	042516	052040
11722	055502	051505	006524	012
11723	055507	127	051117	020104
11724	055514	047516	020123	026461
11725	055522	020064	044507	042526
11726	055530	044040	040505	042504
11727	055536	006522	012	
11728	055541	127	051117	020104
11729	055546	047516	020123	026465
11730	055554	033062	020060	044507
11731	055562	042526	042040	052101
11732	055570	020101	044127	041511
11733	055576	020110	051511	052040
11734	055584	042510	041440	046131
11735	055592	047111	042504	020122
11736	055598	042101	051104	051505
11737	055606	000123		
11738	055630	047506	041522	047111
11739	055636	020107	050117	020111
11740	055644	054502	031440	044440
11741	055652	042116	054105	050040
11742	055660	046125	042523	006523
11743	055666	012		
11744	055667	103	052501	042523
11745	055674	020104	046511	051120
11746	055702	050117	051105	051040
11747	055710	043505	051511	042524
11748	055716	020122	044103	047101
11749	055724	042507	005015	
11750	055730	047507	042117	042040
11751	055736	052101	020101	044507
11752	055744	042526	020123	044127
11753	055752	052101	051440	047510

.ASCII IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER/

EM70: .ASCII READ HEADER AND DATA ERROR IN<(15)<(12)

.ASCII DIFFERENCE LINE TEST<(15)<(12)

.ASCII WORD NOS 1-4 GIVE HEADER<(15)<(12)

.ASCII WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/

EM71: .ASCII FORCING OPI BY 3 INDEX PULSES<(15)<(12)

.ASCII CAUSED IMPROPER REGISTER CHANGE<(15)<(12)

.ASCII GOOD DATA GIVES WHAT SHOULD BE THERE<(15)<(12)

```

1117 0055760 046125 020104 042502
1118 0055766 052040 042510 042522
1119 0055774 005015
1120 0055776 042522 042503 053111
1121 0055780 042105 042040 052101
1122 0055802 020101 044507 042526
1123 0055802 020123 042522 044507
1124 0055806 052123 051105 041440
1125 0055808 047117 042524 052116
1126 0055812 020123 043101 042524
1127 0055816 020122 020063 047111
1128 0055820 042504 020130 052520
1129 0055824 051505 000
1130 0055828 042510 042522 EM72: .ASCII THERE WAS A SETUP ERROR DURING MULTIPLE WRITE/(15)<(12)
1131 0055832 051440 051501 040440
1132 0055836 051440 052105 050123
1133 0055840 042440 051122 051117
1134 0055844 042040 051125 047111
1135 0055848 020107 052515 052114
1136 0055852 050111 042514 052440
1137 0055856 044524 042524 005015
1138 0055860 042510 042101 051105 .ASCII /HEADER AND DATA COMMANDS RESULTING IN AN ABORT/(15)<(12)
1139 0055864 040440 042116 042040
1140 0055868 052101 020101 047503
1141 0055872 046515 047101 051504
1142 0055876 051040 051505 046125
1143 0055880 044524 043516 044440
1144 0055884 020116 047101 040440
1145 0055888 047502 052122 005015
1146 0055892 043117 052040 044510 .ASCII /OF THIS 'OPI' TEST./(15)<(12)<(15)<(12)
1147 0055896 020123 047447 044520
1148 0055900 020047 042524 052123
1149 0055904 006456 006412 012
1150 0055908 124 020117 051124 .ASCII /TO TROUBLE SHOOT SETUP ERROR, LOOP ON THIS TEST/
1151 0055912 052517 046102 020105
1152 0055916 044123 047517 020124
1153 0055920 042523 052524 020120
1154 0055924 051105 047522 026122
1155 0055928 046040 047517 020120
1156 0055932 047117 052040 044510
1157 0055936 020123 042524 052123
1158 0055940 000
1159 0055944 122 040505 020104 EM73: .ASCII /READ HEADER AND DATA FOR 11960 WORDS /(15)<(12)
1160 0055948 042510 042101 051105
1161 0055952 040440 042116 042040
1162 0055956 052101 020101 047506
1163 0055960 020122 030461 033071
1164 0055964 020060 047527 042122
1165 0055968 020123 005015
1166 0055972 044124 052101 044440 .ASCII /THAT IS 46 SECTORS /(15)<(12)
1167 0055976 020123 033064 051440
1168 0055980 041505 047524 051522
1169 0055984 006440 012
1170 0055988 124 040510 020124 .ASCII /THAT IS OVER 3 INDEX PULSES CAUSED AN ERROR

```


11835	056440	051511	047440	042526
11836	056446	020122	020063	047111
11837	056454	042504	020130	052520
11838	056462	051514	051505	041440
11839	056470	052501	042523	020104
11840	056476	047101	042440	051122
11841	056504	051117	000	
11842	056507	122	040505	020104
11843	056514	042510	042101	051105
11844	056522	040440	042116	042040
11845	056530	052101	020101	047506
11846	056536	020122	030461	033071
11847	056544	020060	047527	042122
11848	056552	020123	005015	
11849	056556	044124	052101	044440
11850	056564	020123	033064	051440
11851	056572	041505	047524	051522
11852	056600	020054	044124	052101
11853	056606	044440	020123	053117
11854	056614	051105	031440	044440
11855	056622	042116	054105	026440
11856	056630	012		
11857	056631	120	046125	042523
11858	056636	020123	040503	051525
11859	056644	042105	047440	044520
11860	056652	052040	020117	042523
11861	056650	020124		

EM74: .ASCII /READ HEADER AND DATA FOR 11960 WORDS /'15'<12>

.ASCII /THAT IS 46 SECTORS, THAT IS OVER 3 INDEX /'15'<12>

.ASCIIZ /PULSES CAUSED OPI TO SET/

11862					
11863	056662	040506	040524	020114	CPHALT: .ASCII /FATAL ERROR - SEE DOCUMENT LISTING//15><12>
11864	056670	051105	047522	020122	
11865	056676	020055	042523	020105	
11866	056704	047504	052503	042515	
11867	056712	052116	046040	051511	
11868	056720	044524	043516	005015	
11869	056726	006440	103412	177777	.ASCII / <<15><12><207><377><377><207><377><377><207><377><377>
11870	056734	177607	103777	177777	
11871	056742	044124	020105	047503	.ASCII /THE CONTROLLER OR DEVICE HAS GONE OFFLINE. LOST/<15><12>
11872	056750	052116	047522	046114	
11873	056756	051105	047440	020122	
11874	056764	042504	044526	042503	
11875	056772	044040	051501	043440	
11876	057000	047117	020105	043117	
11877	057006	046106	047111	026105	
11878	057014	046040	051517	006524	
11879	057022	012			
11880	057023	047	042522	042101	.ASCII /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
11881	057030	023531	020054	042502	
11882	057036	047503	042515	052440	
11883	057044	040516	040526	046111	
11884	057052	041101	042514	020054	
11885	057060	051117	044040	051501	
11886	057066	051440	040524	052524	
11887	057074	020123	044502	051524	
11888	057102	005015			
11889	057104	044127	041511	020110	.ASCIZ /WHICH CANNOT BE CLEARED/
11890	057112	040503	047116	052117	
11891	057120	041040	020105	046103	
11892	057126	040505	042522	000104	
11893					
11894					
11895					
11896	057134	041520	020040	020040	DH1: .ASCII /PC TEST WAIT BIT REG REG RHCS1/<15><12>
11897	057142	020040	042524	052123	
11898	057150	020040	020040	040527	
11899	057156	052111	020040	020040	
11900	057164	044502	020124	020040	
11901	057172	020040	042522	020107	
11902	057200	020040	020040	042522	
11903	057206	020107	020040	020040	
11904	057214	044122	051503	006461	
11905	057222	012			
11906	057223	040	020040	020040	.ASCIZ / NO PC EXPCTD ADDRESS CONTENT CONTENT /
11907	057230	020040	047040	020117	
11908	057236	020040	020040	050040	
11909	057244	020103	020040	020040	
11910	057252	042440	050130	052103	
11911	057260	020104	040440	042104	
11912	057266	042522	051523	041440	
11913	057274	047117	042524	052116	
11914	057302	041440	047117	042524	
11915	057310	052116	000011		

12024												
12025	060424	041520	020040	020040	DH51:	.ASCII	/PC	TEST	REG	GOOD	RECVD	ILLEGL/<15><12>
12026	060432	020040	042524	052123								
12027	060440	020040	020040	042522								
12028	060446	020107	020040	020040								
12029	060454	047507	042117	020040								
12030	060462	020040	042522	053103								
12031	060470	020104	020040	046111								
12032	060476	042514	046107	005015								
12033	060504	020040	020040	020040		.ASCIZ	/	NO	ADDRESS	DATA	DATA	FUNCTN/
12034	060512	020040	047516	020040								
12035	060520	020040	020040	042101								
12036	060526	051104	051505	020123								
12037	060534	040504	040524	020040								
12038	060542	020040	040504	040524								
12039	060550	020040	020040	052506								
12040	060556	041516	047124	000								
12041												
12042	060563	120	020103	020040	DH60:	.ASCII	/PC	TEST	REG	GOOD	RECVD	MODFING/<15><12>
12043	060570	020040	052040	051505								
12044	060576	020124	020040	051040								
12045	060604	043505	020040	020040								
12046	060612	043440	047517	020104								
12047	060620	020040	051040	041505								
12048	060626	042126	020040	046440								
12049	060634	042117	044506	043516								
12050	060642	005015										
12051	060644	020040	020040	020040		.ASCIZ	/	NO	ADDRESS	DATA	DATA	REG/
12052	060652	020040	047516	020040								
12053	060660	020040	020040	042101								
12054	060666	051104	051505	020123								
12055	060674	040504	040524	020040								
12056	060702	020040	040504	040524								
12057	060710	020040	020040	042522								
12058	060716	000107										
12059	060720	041520	020040	020040	DH61:	.ASCII	/PC	TEST	PC OF	RHDS1/<15><12>		
12060	060726	020040	042524	052123								
12061	060734	020040	020040	041520								
12062	060742	047440	004506	051040								
12063	060750	042110	030523	005015								
12064	060756	020040	020040	020040		.ASCIZ	/	NO	JSR	WAS/		
12065	060764	020040	047516	020040								
12066	060772	020040	020040	051512								
12067	061000	020122	020040	020040								
12068	061006	040527	000123									
12069	061012	041520	020040	020040	DH62:	.ASCII	/PC	PC OF	RHCS1/<15><12>			
12070	061020	020040	041520	047440								
12071	061026	020106	020040	044122								
12072	061034	051503	006461	012								
12073	061041	040	020040	020040		.ASCIZ	/	JSR	WAS/			
12074	061046	020040	045040	051123								
12075	061054	020040	020040	053440								
12076	061062	051501	000									
12077	061065	120	020103	020040	DH65:	.ASCII	/PC	TEST	CONT	CONT	CONT/<15><12>	

12132	061512	002262	002260	002304			
12133	061520	002264	000000				
12134	061524	001116	004504	002262	DT26:	.WORD	\$ERRPC, TSTNM, CS1, CS2, DS1, FR1, ER2, ER3, 0
12135	061532	002260	002304	002264			
12136	061540	002270	002276	000000			
12137	061546	001116	004504	004502	DT30:	.WORD	\$ERRPC, TSTNM, ERWORD, \$GDDAT, \$BDDAT, 0
12138	061554	001124	001126	000000			
12139	061562	001116	004504	004500	DT51:	.WORD	\$ERRPC, TSTNM, REGADR, \$GDDAT, \$BDDAT, ILLEGL, 0
12140	061570	001124	001126	002364			
12141	061576	000000					
12142	061600	001116	004504	004500	DT60:	.WORD	\$ERRPC, TSTNM, REGADR, \$GDDAT, \$BDDAT, \$BDADR, 0
12143	061606	001124	001126	001122			
12144	061614	000000					
12145	061616	001116	004504	033040	DT61:	.WORD	\$ERRPC, TSTNM, PCJSR, \$BDADR, 0
12146	061624	001122	000000				
12147	061630	001116	004504	033040	DT62:	.WORD	\$ERRPC, TSTNM, PCJSR, \$BDADR, 0
12148	061636	001122	000000				
12149	061642	001116	004504	002262	DT65:	.WORD	\$ERRPC, TSTNM, CS1, AS, DS1, 0
12150	061650	002300	002304	000000			
12151	061656	001116	004504	002266	DT66:	.WORD	\$ERRPC, TSTNM, DST, ER1, ER2, ER3, CS1, CS2, 0
12152	061664	002264	002270	002276			
12153	061672	002262	002260	000000			
12154	061700	001116	004504	002262	DT72:	.WORD	\$ERRPC, TSTNM, CS1, CS2, DS1, DST, CA, ER1, WC, 0
12155	061706	002260	002304	002266			
12156	061714	002274	002264	002254			
12157	061722	000000					
12158							
12159	061724	000	000	000	DF1:	.BYTE	0,0,0,0,0,0,0
12160	061727	000	000	000			
12161	061732	000					
12162	061733	000	000	000	DF4:	.BYTE	0,0,0,0,0,1,0
12163	061736	000	000	001			
12164	061741	000					
12165	061742	000	000	000	DF5:	.BYTE	0,0,0,0,0
12166	061745	000	000				
12167	061747	000	000	000	DF6:	.BYTE	0,0,0,0
12168	061752	000					
12169	061753	000	000	000	DF7:	.BYTE	0,0,0
12170	061756	000	000	000	DF10:	.BYTE	0,0,0,0,0,0,0
12171	061761	000	000	000			
12172	061764	000					
12173							
12174	061765	000	000	000	DF26:	.BYTE	0,0,0,0,0,0,0,0
12175	061770	000	000	000			
12176	061773	000	000				
12177							
12178	061775	000	000	000	DF30:	.BYTE	0,0,0,0,0
12179	062000	000	000				
12180							
12181	062002	000	000	000	DF51:	.BYTE	0,0,0,0,0,0,0
12182	062005	000	000	000			
12183							
12184	062010	000	000	000	DF60:	.BYTE	0,0,0,0,0,0,0
12185	062013	000	000	000			

12196	062016	000	000	000	DF61:	.BYTE	0,0,0,0
12197	062021	000					
12198	062022	000	000	000	DF62:	.BYTE	0,0,0,0
12199	062025	000					
12190	062026	000	000	000	DF65:	.BYTE	0,0,0,0,0
12191	062031	000	000				
12192	062033	000	000	000	DF66:	.BYTE	0,0,0,0,0,0,0,0
12193	062036	000	000	000			
12194	052041	000	000	000			
12195							
12196	062044	000	000	000	DF72:	.BYTE	0,0,0,0,0,0,0,0
12197	052047	000	000	000			
12198	062052	000	000	000			
12199							
12200	062056					.EVEN	
12201							
12202	000001				.END		



ERRFLGS = 004632	1392*	7787*	10277*															
ERR = 040000	1087*	3465	3807	3951	4725	4869	5054	5193	5552	5804	6262	6553	6870					
ERRVEC = 000004	7167*	7678	7848	8402	8518													
ERRWORD = 004502	159*	1455	1456*	1467*	1592*	1600*	1623*	9799	9800*	9802*	9805*							
ERR1 = 002264	1353*	9681*	9687*	12137														
ERR2 = 002270	1296*	8229*	8229	10378	12131	12134	12151	12154										
ERR3 = 002276	1298*	10386	12134	12151														
EXT1 = 000001	1301*	10394	12134	12151														
EXT10 = 000010	1150*																	
EXT10 = 000010	1153*																	
EXT20 = 000020	1151*																	
EXT20 = 000020	1154*																	
EXT4 = 000004	1152*																	
EXT40 = 000040	1155*																	
FEN = 000200	1176*																	
FER = 000020	1095*	6278	6565															
FILL = 033740	9452*																	
FILLRE = 032706	2351	2355	2375	2507	2512	2531	2762	2767	2772	2917	2922	2927	2932					
	3163	3168	3173	3278	3183	3349	3354	3359	3483	3628	3633	3638	3733					
	3738	3755	3760	3784	3927	3932	3953	4350	4355	4360	4499	4504	4509					
	5512	5517	5540	5569	5574	5581	5586	5778	5783	5808	5821	5826	5831					
	5984	5989	6212	6217	6227	6232	6264	6385	6390	6395	6509	6514	6519					
	6888	7180	7185	7190	7323	7329	7334	7494	7499	7504	7509	7514	7519					
	7850	9088*																
FINACC = 004564	1365*	9239*																
FINAL = 004562	1364*	9240*																
FIRE = 004634	1394*	1479	1516*															
FLHRC = 032630	2233	2245	22414	2636	2654	2901	3147	3272	3550	3839	4231	4253	4595					
	4596	5267	5353	5365	5397	5409	5639	5651	5945	6072	6653	6993	7001					
	7931	8065	8324	8447	8650	8785	9031*											
FMT22 = 010000	1197*	2284	2285	2286	2304	2305	2306	2443	2444	2445	2462	2463	2464					
	2690	2691	2692	2824	2825	2826	3043	3044	3045	3302	3303	3304	3305					
	3413	3414	3580	3581	3582	3684	3685	3686	3880	3881	3882	3883	3884					
	4281	4428	4429	4430	4635	4636	4637	4654	4655	4656	4892	4893	4894					

CPERSE 034140
CP = 020000
CR = 000200
PAR = 000010
PC = 000007

9552*	10020	10103											
1104*	7860	8229											
1033*	3748	3770	5536	6253	6280	6281	6400	6401	6571	6572	7618	7628	
7878	7879												
1094*													
90*	1478*	1669*	1814*	1954*	1996*	2041*	2084*	2090*	2094*	2183*	2219*	2322*	
2392*	2407*	2477*	2548*	2568*	2596*	2601*	2629*	2705*	2730*	2810*	2852*	2950*	
2999*	2992*	2998*	3029*	3060*	3202*	3221*	3265*	3318*	3376*	3330*	3427*	3503*	
3522*	3543*	3597*	3654*	3699*	3820*	3832*	3896*	3982*	4002*	4033*	4039*	4070*	
4092*	4178*	4224*	4294*	4379*	4399*	4405*	4442*	4528*	4548*	4569*	4572*	4756*	
4778*	4817*	4997*	4922*	4929*	4962*	5002*	5082*	5107*	5146*	5219*	5260*	5305*	
5322*	5482*	5508*	5626*	5686*	5748*	5854*	5874*	5882*	5918*	5949*	6006*	6026*	
6064*	6107*	6136*	6179*	6298*	6309*	6355*	6419*	6440*	6459*	6506*	6589*	6609*	
6641*	6653*	6692*	6723*	6806*	6911*	6929*	6962*	6975*	7036*	7109*	7212*	7222*	
7267*	7352*	7368*	7389*	7432*	7527*	7543*	7561*	7596*	7647*	7731*	7759*	7781*	
7895*	7901*	7920*	7959*	7997*	8007*	8054*	8092*	8131*	8144*	8171*	8195*	8231*	
8241*	8272*	8286*	8315*	8361*	8400*	8426*	8484*	8515*	8545*	8586*	8590*	8642*	
8587*	8746*	8820*	8868*	8932*	8995*	9005*	9010	9143*	9187*	9190*	9502*	9512*	
9623*	9688*	9717*	9946*	9953*	9960*	9974*	9976*	10004*	10019*	10073*	10124*	10292*	
10537*	10691												
9159*	9161*	9162*	9167*	9168*	12145	12147							
1279*	9263*												
1278*	9237*	9264*											
1280*	9238												
1036*													
1086*													
76*													
170*													
1336*	2098	2119	2140										
1182*													
1222*													
1558*	8959*	9554*	10323*	10485*	10503	10507*							
1082*													
93*													
94*													
95*													
96*													
97*													
99*													
99*													
100*													
73*	74	1471*	2005*	2050*	2085*	2872*	3121*	8933*	9553*				
1065*													
1219*													
74*													
7997	8131	8231	8241	8400	8516	9502	9610*						
165*	1446*	1447*	10659*	10660*	10669*	10675*	10687*	10689*					
10178	10651*												
10225	10652*												
1568	9580	9591	9718	9735	10653*								
1062*	2006	2019	2051	2141	2499	2623	2731	2894	3073	3115	3140	3342	
3451	3621	3723	3920	4125	4319	4468	5328	5505	5771	5972	6130	6302	
6378	6529	6716	6837	7060	7139	7292	7463	7657	7985	8119	8220	8391	
8507	8715	8848	9173										

PCJSR 033040
PCLBUE 002246
PCLCSR 002244
PCLCTR 002250
PGE = 002000
PIP = 020000
PIRQ = 177772
PIRQVE = 000240
PKACK = 002360
PLU = 020000
PRE = 000020
PRITEM 040434
PROG = 001000
PRO = 000000
PR1 = 000040
PR2 = 000100
PR3 = 000140
PR4 = 000200
PR5 = 000240
PR6 = 000300
PR7 = 000340
PS = 177776
PSEL = 002000
PSU = 000001
PSW = 177776
PUTREG 034726
PWARVE = 000024
RDCHR = 104407
RDLIN = 104410
RDOC = 104411
RDY = 000200

		5984*	5989*	5996*	6027*	6072*	6082*	6090*	6146*	6154*	6172*	6212*	6217*	6227*
		6232*	6239*	6248*	6255*	6264*	6273*	6288*	6315*	6323*	6331*	6348*	6385*	6390*
		6395*	6409*	6431*	6466*	6474*	6482*	6499*	6537*	6546*	6555*	6560*	6579*	6599*
		6658*	6669*	6675*	6729*	6760*	6766*	6774*	6781*	6799*	6826*	6827*	6829	6844*
		6851*	6858*	6863*	6876*	6883*	6888*	6896*	6906	6907*	6908	6910*	6920*	6983*
		6993*	7001*	7011*	7019*	7069*	7075*	7083*	7101*	7128*	7129*	7131	7146*	7153*
		7160*	7173*	7180*	7185*	7190*	7198*	7208	7209*	7210	7212*	7229*	7235*	7243*
		7260*	7286*	7303	7306	7323*	7329*	7334*	7342*	7359*	7396*	7403*	7410*	7426*
		7453*	7474	7477	7494*	7499*	7504*	7509*	7517*	7534*	7574*	7576	7640*	7664*
		7671*	7687*	7698*	7708*	7714*	7721*	7766*	7767	7769*	7771*	7772*	7773*	7777*
		7792*	7804*	7805*	7806*	7807*	7818*	7819*	7822*	7823*	7826*	7827*	7832*	7941*
		7850*	7855*	7866*	7885*	7931*	7942*	7978*	8065*	8076*	8112*	8179*	8212*	8324*
		8335*	8344*	8437*	8447*	8458*	8467*	8535*	8650*	8661*	8670*	8776*	8785*	8794*
		8903*	8859*	8962*	8964	9002*	9005	9034	9035	9039	9044*	9064	9065	9066
		9073*	9091	9092	9097*	9108	9109	9112*	9126	9128*	9135	9137*	9206	9207
		9208	9215*	9265	9269*	9270	9272	9273	9274	9275	9276	9329*	9352	9355*
		9356	9358	9359	9360	9361	9409*	9425	9426	9428	9430	9432	9437*	9457
		9458	9459	9460	9471*	9495	9497	9498	9499	9500*	9528*	9611	9614*	9617
		9622*	9637	9638	9639	9640	9642	9644	9646	9647*	9674	9675	9676	9677
		9678*	9702*	9719*	9721*	9722*	9854	9864*	9868	9884	9885	9898*	9929	9930*
		9931	9934*	10222	10226*	10227	10230	10250*	10253*	10489	10490*	10491*	10498*	10499*
		10500*	10501*	10502*	10503	10505	10507	10508	10513	10519	10521*	10522	10536*	10626
		10627*	10628	10629*	10630*	10631*	10632*	10661	10686*					
R1	=%.000001	82*	1595*	1596	1602	1636*	1637*	1638	1642	1644	1677*	1686	1699	1731
		2006*	2019	2051*	2126	2147	2724	2746	3083	3085	3087	3131*	4059	4110
		4131	4313	4334	4462	4483	7286	7307	7453	7478	7653*	7785*	7978	8112
		8212	8963*	8964	8967	9032	9034*	9039*	9043*	9061	9064*	9067*	9072*	9089
		9091*	9093*	9094*	9096*	9140*	9147*	9171	9176	9203	9206*	9209	9214*	9256
		9274*	9309*	9310	9319*	9320	9328*	9423	9425*	9427*	9430*	9432*	9436*	9453
		9457*	9463*	9470*	9491	9496*	9507	9509	9514	9527*	9612	9615*	9617*	9621*
		9669	9674*	9682	9685	9689	9701*	9720*	9723*	9855	9868*	9869	9873	9997*
		10223	10228*	10236*	10238*	10240*	10243*	10246	10249*	10518	10519*	10524	10529	10530
		10535*	10662	10685*										
R2	=%.000002	83*	1596*	1678*	1679*	1683*	2013*	2025*	2057*	2065*	9033	9035*	9040*	9042*
		9062	9065*	9068*	9071*	9090	9092*	9094	9095*	9141*	9145*	9146*	9204	9207*
		9209*	9213*	9267	9275*	9309	9318*	9319	9327*	9424	9426*	9433*	9435*	9454
		9458*	9465*	9469*	9492	9497*	9507	9510	9514	9526*	9613	9616*	9618*	9620*
		9670	9675*	9682	9686	9689	9700*	9856	9867*	9871*	9874	9881*	9882*	9883
		9889*	9896*	10224	10229*	10237*	10239*	10241*	10247	10248*	10663	10684*		
R3	=%.000003	84*	1723*	1724*	1727*	1772*	1955	1971	2127	2156	2725	2755	3097	3099
		3101	4059	4111	4140	4314	4343	4463	4492	7287	7316	7454	7487	7979
		8113	8213	9063	9066*	9067	9070*	9142*	9163	9169	9183	9205	9208*	9210*
		9212*	9268	9280*	9283*	9285*	9288*	9326*	9353	9365*	9368*	9370*	9373*	9394*
		9395*	9408*	9455	9459*	9463	9464*	9468*	9493	9498*	9519*	9525*	9671	9676*
		9679	9687	9694*	9699*	9857	9865*	9866*	9880*	9883*	9892*	9893*	9895*	10174
		10175*	10176	10179*	10180	10184	10186	10188*	10190*	10574	10583*	10589*	10590*	10593*
		10598*	10599*	10600	10609*	10664	10683*							
R4	=%.000004	85*	1681*	1682*	9143*	9456	9460*	9464	9467*	9494	9499*	9512	9524*	9672
		9679*	9680*	9681	9698*	10575	10577*	10578*	10579*	10580	10581*	10595	10597*	10605*
		10608*	10665	10682*										
R5	=%.000005	86*	1728*	1735	1758	1772	1774	1778*	2127*	2152	2155	2725*	2751	2754
		3074*	3097	3100	4059*	4111*	4136	4139	4314*	4339	4342	4463*	4488	4491
		7287*	7312	7315	7454*	7483	7486	7575*	7590*	7774*	7778*	7979*	8011*	8012*
		8013	8015	8016	8113*	8148*	8149*	8150	8152	8153	8213*	9495	9504*	9506*

		9511	9523*	9673	9677*	9698	9697*	9858	9860*	9862*	9869*	9873*	9898	9894*
		10576	10582*	10594*	10586*	10587*	10588*	10589	10607*	10666	10681*			
R6	=:007006	87*	89	1434*	1435*	1436								
R7	=:000007	88*	90											
SAVER	033152	2104	2315	2471	2698	2833	3052	3311	3420	3590	3692	3889	4084	4287
		4436	4666	4810	4996	5139	5475	5741	5943	6172	6348	6499	6799	7101
		7260	7426	7640	7792	9202*								
SAVERE	004512	1363*	2106	2175	2317	2384	2473	2540	2700	2782	2835	2943	3054	3194
		3313	3369	3422	3488*	3496	3592	3647	3694	3789*	3813	3891	3965*	3974
		4086	4161*	4169	4289	4371	4438	4520	4668	4709*	4711*	4713*	4715*	4734*
		4737*	4748	4812	4853*	4855*	4857*	4859*	4870*	4880*	4889	4998	5038*	5040*
		5042*	5044*	5055*	5065*	5074	5141	5184*	5202*	5211	5477	5556*	5594*	5600
		5743	5836*	5839*	5846	5945	5998	6174	6269*	6281*	6282*	6290	6350	6401*
		6402*	6411	6501	6567*	6572*	6573*	6581	6901	6872*	6898	7103	7159*	7200
		7262	7344	7428	7519	7642	7680*	7693*	7723	7794	7862*	7875*	7876*	7879*
		7880*	7887	9094*	9430*	9432*	9503							
SC	= 100000	1069*	3472	3780	3963	4152	5181	5527	5793	6244	6542	6956	7153	7605
		7615	7625	7634	7692	7837	7995	8129	9239					
SC1	= 000100	1156*												
SC10	= 001000	1159*												
SC2	= 000200	1157*												
SC20	= 002000	1160*												
SC4	= 000400	1158*												
SEECOM	002352	1333*	3013	4947	6741	9127								
SEEKCY	032754	3009	4942	6729	9126*									
SELECT	004626	1387*	1421*	1423*	1426*	1560	1787	1825	8948					
SELTST	007616	1695	1787*											
SEARCH	002334	1326*	2827	2857	3046	3071	3078	4105	4124	9110				
SKI	= 040000	1225*												
SN	002310	1306*	1936*	10482										
SN01	005774	1482	1516*											
SP	=:000006	89*	1438*	1455*	1463*	1467	1569*	1570	1571	1589*	1601	1616*	1634*	1758*
		1764*	1774*	1811*	1878*	1884*	1890*	1994*	2018	2039*	2062	2087*	2119*	2120*
		2122	2140*	2141*	2142	2143	2150*	2151	2152	2216*	2333*	2334*	2336	2498*
		2489*	2491	2593*	2614*	2615*	2617	2718*	2719*	2721	2739*	2740*	2741	2742
		2749*	2750	2751	2843*	2844*	2845	2857*	2858*	2860	2989*	3013*	3014*	3016
		3078*	3079*	3081	3262*	3331*	3332*	3334	3440*	3441*	3443	3540*	3610*	3611*
		3613	3712*	3713*	3715	3909*	3910*	3912	4030*	4052*	4053*	4055	4105*	4106*
		4109	4124*	4125*	4126	4127	4134*	4135	4136	4221*	4307*	4308*	4310	4327*
		4328*	4329	4330	4337*	4338	4339	4456*	4457*	4459	4476*	4477*	4478	4479
		4486*	4487	4488	4566*	4686*	4687*	4689	4775*	4830*	4831*	4833	4919*	4947*
		4948*	4950	5015*	5016*	5018	5104*	5159*	5160*	5162	5257*	5319*	5319*	5321
		5495*	5496*	5498	5761*	5762*	5764	5962*	5963*	5965	6061*	6120*	6121*	6123
		6192*	6193*	6195	6279*	6280*	6282	6368*	6369*	6371	6399*	6400*	6402	6455*
		6519*	6520*	6522	6570*	6571*	6573	6638*	6705*	6706*	6708	6741*	6742*	6744
		6819*	6820*	6822	6906*	6910	6960*	7049*	7050*	7052	7121*	7122*	7124	7208*
		7212	7280*	7281*	7283	7300*	7301*	7302	7303	7310*	7311	7312	7366*	7447*
		7448*	7450	7471*	7472*	7473	7474	7481*	7482	7483	7558*	7651*	7552*	7653
		7756*	7877*	7879*	7880	7917*	7972*	7973*	7975	8051*	8106*	8107*	8109	8206*
		8207*	8209	8269*	8299*	8300*	8302	8374*	8375*	8377	8497*	8498*	8500	8533*
		8611*	8612*	8614	8700*	8701*	8703	8759*	8760*	8762	8838*	8839*	8841	8893*
		8894*	8895	8938*	8944*	8954*	8999*	9032*	9033*	9042	9043	9061*	9062*	9063*
		9070	9071	9072	9089*	9090*	9095	9096	9161	9163*	9164*	9167	9169*	9171*
		9172*	9173	9180*	9181	9199*	9203*	9204*	9205*	9212	9213	9214	9265*	9266*

9267*	9268*	9269	9276*	9326	9327	9328	9329	9352*	9353*	9355	9361*	9409
9409	9423*	9424*	9435	9436	9453*	9454*	9455*	9456*	9467	9469	9469	9470
9491*	9492*	9493*	9494*	9495*	9515*	9516*	9517	9523	9524	9525	9526	9527
9559*	9565*	9581*	9582	9592	9593*	9611*	9612*	9613*	9620	9621	9622	9641*
9642*	9643	9669*	9670*	9671*	9672*	9673*	9690*	9691*	9692	9697	9698	9699
9700	9701	9711*	9722	9729*	9736	9745*	9751*	9793*	9802	9804	9805	9833
9834	9838*	9854*	9855*	9856*	9857*	9858*	9859*	9860	9863*	9876	9878*	9880
9890	9892	9894	9895	9896	9897	9898	9900*	9901*	9929*	9930	9931*	9933
9934	9935*	9937	9939	9941	9947	9949*	9951*	9959*	9963	9967	9968	9972
10013*	10014*	10015	10019	10021	10030	10032	10035	10039*	10039*	10040	10045	10047
10049*	10050	10066*	10067*	10068	10078	10085*	10098*	10089*	10093*	10094*	10096	10099*
10106	10109*	10113	10115	10117	10118*	10125	10127	10129*	10130	10132*	10133*	10134*
10135*	10136*	10151*	10152*	10153*	10154*	10155*	10161*	10174*	10179	10190	10191*	10192*
10193*	10220*	10221*	10222*	10223*	10224*	10226	10230*	10232	10234	10242*	10243	10245
10246*	10248	10249	10250	10252	10287	10301*	10304*	10325*	10326*	10327	10338*	10346*
10354*	10362*	10370*	10378*	10386*	10394*	10402*	10410*	10418*	10426*	10434*	10442*	10450*
10458*	10456*	10474*	10482*	10489*	10494*	10518*	10524*	10529*	10535	10536	10556*	10567
10568	10569*	10574*	10575*	10576*	10582	10607	10608	10609	10610*	10611*	10626*	10627
10661*	10662*	10663*	10664*	10665*	10666*	10667*	10668	10676*	10680	10681	10682	10683
10684	10685	10686	10691*									
3066	4077	9108*										
SW04	032734											
SW07	177572	218*										
SW11	177574	219*										
SW21	177576	220*										
SW22	172516	221*										
SWT1	031044	8432*										
SWT2	031052	8433*										
SWT3	031072	8421*	8442	8565								
SWT4	031112	8422*	8445*	8556*								
SWT5	031124	8423*	9456*	8557*								
STACK	001000	8423*	8465*	9558*								
		64*			1634	1811	1994	2039	2097	2216	2593	2989
		4030	1438	1589	4775	4919	5104	5257	6061	6456	6638	6960
		7756	4221	4566	8269	8583						
		1422	7917	8051								
			1425	1430*								
		75*										
START	004720											
STK LMT	177774											
ST22	024066	6651*	6934									
ST22A	022464	5994*										
ST23	024644	6924	6931*									
ST24	024716	6973*	7374									
ST28	025732	7364	7371*									
SWR	001140	274*	1436	1457*	1459	1465*	1521	1528	1604	1671	1693	9515
		9808	9810	9816	9823	10023	10062	10117*	10283	10290	10295	10299
		10680*										9630
		180*	1465	1521	10023	10062	10085					10325
		128*										9794
SW0	000001	118*	128									10667
SW00	000001	117*	127									
SW01	000002	116*	126									
SW02	000004	115*	125									
SW03	000010	114*	124									
SW04	000020	113*	123									
SW05	000040	112*	122	10327								
SW06	000100	111*	121	9517	9692							
SW07	000200	110*	120									
SW08	000400											

SW04 = 032734
SW07 = 177572
SW11 = 177574
SW21 = 177576
SW22 = 172516
SWT1 = 031044
SWT2 = 031052
SWT3 = 031072
SWT4 = 031112
SWT5 = 031124
STACK = 001000
START = 004720
STK LMT = 177774
ST22 = 024066
ST22A = 022464
ST23 = 024644
ST24 = 024716
ST28 = 025732
SWR = 001140
SW0 = 000176
SW0 = 000001
SW00 = 000001
SW01 = 000002
SW02 = 000004
SW03 = 000010
SW04 = 000020
SW05 = 000040
SW06 = 000100
SW07 = 000200
SW08 = 000400

STMP5 = 001210
STN = 000041

297*	12	1561	1572	1578	1585*	1586	1526	1633*	1639	1649	1653*	1665
1788	1794	1808*	1811	1939	1948*	1956	1981	1990*	1991	2014	2020	2029
2035*	2036	2058	2071	2080*	2087	2202	2216*	2575	2533*	2974	2989*	3250
3262*	3525	3540*	4019	4030*	4187	4221*	4556	4566*	4765	4775*	4906	4919*
5094	5104*	5231	5257*	6048	6061*	6445	6456*	6618	6638*	6933	6936*	6950*
7373	7377	7386*	7547	7558*	7568	7746	7756*	7903	7917*	8030	8051*	8135
8249	8269*	8562	8567	8583*	8921	8932*						

STYPR = 001152
STYPR 6 = 001157
STYPS = 001150
STYPP = 042012
STYRP = 000013

279*	9967*	9978										
283*	9925	9978										
278*	9965	9978										
1444	10626*											
10634*	10643*	10644*	10645*	10646*	10647*	10648	10649*	10650	10651*	10652*	10653*	10654*
10655*												

STRPAD 042034
STSYXN 001102
STSYIN 040054
STYBZ = *****
STYBJS 036342
STYBMS 036566
STYBMC 036736
STYBXC 037004
STYBON 041610
STYBOS 041624
STYBOS 041554
XTSTR 036102
STYBNT 000000
STYBNT 042007
STYBNT *****
STYBNT 062056

10631	10641*											
256*	8947*	8988*	9783	9810	9832*	9837	9841	10282	10307			
10175	10176	10193	10197*									
10547*												
9853*	10646											
9925*	10634	10542										
9946	9953	9960	9965*	9966	10124							
9971	9973	9976*										
10571*	10643											
10570	10573*	10645										
10565*	10644											
9797*												
9004*												
10567*	10571*	10581	10616*									
9794	10292											

174*	178*	186	187*	189*	191*	195*	197*	199*	246*	253*	304	1344*
1345*	1346*	1363*	1379*	1437	1451	1452	1492*	1501*	1505*	1676*	1702*	1705*
1714*	1757*	1763*	1769*	1847*	1877*	1916*	1922*	1928*	1966*	1970*	1976*	8937*
8943*	9012	9016	9571*	9579*	9586*	9728*	9768*	9840	9841	9907*	9978	9991
9985*	9986	9987*	10196*	10198	10205	10259	10306	10327*	10345*	10409*	10417*	10425*
10733*	10441*	10449*	10473*	10481*	10539*	10571	10695	12200*				

H10

MSG	1578# 2073 4555# 6620 8556#	1590 2201# 4558 6936# 8569	1625# 2204 4764# 6938 8921#	1628 2574# 4767 7376# 8923	1648# 2577 4905# 7379 8923	1651 2973# 4908 7546#	1793# 2976 5093# 7549	1796 3249# 5096 7745#	1939# 3252 5230# 7748	1941 3525# 5233 7902#	1980# 3527 6047# 7905	1983 4018# 6050 8029#	2027# 4021 6445# 8032	2030 4187# 6447 8248#	2070# 4189 6617# 8251	
MULT NEWST	171# 171# 4187 8249	1578 4556 8567	1626 4765 8921	1649 4906	1794 5094	1939 5231	1981 6048	2028 6445	2071 6618	2202 6936	2575 7377	2974 7547	3250 7746	3525 7903	4019 8030	
OFFST POP	46# 171# 10249	6910 10690	7212 10681	9042 9070	9070 9095	9212 9095	9325 9212	9407 9325	9435 9407	9467 9435	9522 9467	9620 9522	9620 9620	9696 9696	9994 9994	
PUSH	171# 10222	6905 10661	7207 10667	9031	9060	9088	9202	9265	9351	9422	9452	9490	9610	9668	9853	
REPORT RFORGC RHOLEA	171# 46# 46# 3830 5917 8005	1813 4032 6063 8053	1995 4069 6135 8142	2040 4223 6307 8170	2089 4403 6458 8271	2218 4568 6640 8314	2405 4777 6651 8425	2595 4921 6962 8585	2628 4962 6973 8641	2808 5106 7220 8744	2991 5259 7388	3028 5332 7550	3264 5624 7594	3398 5685 7758	3542 5880 7919	
SAVE SAVTST SCH SCOPE	46# 46# 46# 66# 4220 8268	10275 1586 3065 1594 4565 8582	1665 4076 1632 4774 8931 4941	1948 1991 2036	1991 2036	1947 1989 2034 2079 2215 2592 2998 3261 3539 4029	1989 2034 2079 2215 2592 2998 3261 3539 4029	2034 2079 2215 2592 2998 3261 3539 4029	2079 2215 2592 2998 3261 3539 4029	2215 2592 2998 3261 3539 4029	2592 2998 3261 3539 4029	2998 3261 3539 4029	3261 3539 4029	3539 4029	4029	
SEEKCO SETPRI SETTRA SETUP SKIP SLASH SPACE SREGIS	46# 171# 10634# 171# 46# 171# 171# 46# 4666 7792	10154 10643 1432 171#	10644 10645 10646 10648 10650 10651 10652 10653 10654	10645 10646 10648 10650 10651 10652 10653 10654	10646 10648 10650 10651 10652 10653 10654	10648 10650 10651 10652 10653 10654	10650 10651 10652 10653 10654	10651 10652 10653 10654	10652 10653 10654	10653 10654	10654	10654	10654	10654	10654	10654
STARS	38 1631 2028 2987 4617 5390 6059 8030 8930 10620	44 1649 2033 3229 4621 5394 6445 8049 8980 10657	171# 1661 2071 3243 4743 5430 6454 8249 9763 10673	184 1741 2078 3250 4765 5434 6618 8267 9772	249 1751 2202 3260 4773 5562 6636 8567 9780	304 1794 2214 3525 4906 5566 6936 8581 9843	1005 1806 2226 3538 4917 5566 6958 8581 9910	1007 1859 2230 4019 4917 5623 6958 8629 9910	1052 1870 2268 4019 5102 5632 7377 8633 9980	1054 1896 2271 4028 5231 5636 7547 8738 10057	1532 1929 2428 4187 5255 5695 7556 8879 10072	1554 1939 2431 4556 5286 5699 7746 8893 10143	1578 1946 2575 4564 5343 5815 7754 9899 10167	1583 1981 2591 4578 5346 5831 7903 9903 10167	1626 1988 2974 4582 5353 5849 7915 8921 10543	
STARTT	46# 6061	1811 6456	2087 6638	2216 6960	2593 7386	2989 7558	3262 7756	3540 7917	4030 9051	4221 8269	4566 8583	4775	4919	5104	5257	
SWRSU TJUMP TRMTRP TSCLR2 TSCLR5 TTSTNO	171# 46# 10634# 46# 46# 46#	1452# 6933	7373	7568	8135	8135	8135	8135	8135	8135	8135	8135	8135	8135	8135	
	46#	1812	2088	2217	2594	2990	3263	3541	4031	4222	4567	4776	4920	5105	5258	

ADD	7578	7597	8413	8414	8415	8556	8557	8558	8891	8912	9189	9318	9319	9464	9506
	9581	9873	9935	10099	10109	10118	10243	10502	10569	10579					
ASL	10132	10133	10134	10236	10238	10240	10499	10500	10501	10630					
ASLB	9878														
BCC	1733	9879													
BEC	1529	1552	1573	1640	1690	1694	1737	1739	1743	1745	1748	1750	1789	1863	1855
	1898	1900	1903	1905	1908	1910	1972	2021	2145	2154	2228	2270	2430	2744	2753
	3084	3098	3730	4129	4138	4332	4341	4481	4490	4580	4619	5348	5392	5432	5564
	5634	5697	5817	6209	7305	7314	7476	7485	7567	7996	8014	8130	8151	8230	8240
	8405	8521	8631	8881	8901	8906	8909	8949	8961	8965	9003	9175	9182	9292	9304
	9377	9400	9429	9508	9518	9683	9693	9809	9811	9913	9817	9826	9938	9973	10024
	10077	10116	10131	10159	10231	10281	10284	10300	10303	10509	10514	10520	10531	10596	
BGE	9829														
BGT	8994	9987	10048	10128	10233	10603									
BHI	9815														
BHIS	7604	9313	9322												
BIC	1569	6280	6281	6400	6401	6571	6572	7807	7819	7823	7827	7878	7979	8991	9172
	9180	9432	9516	9691	9721	10014	10039	10049	10067	10094	10129	10242	10326	10593	
BIS	2120	2141	2334	2489	2615	2719	2740	2844	2858	3014	3073	3079	3108	3332	3441
	3488	3511	3713	3789	3910	3955	4053	4106	4125	4161	4308	4328	4457	4477	4687
	4734	4831	4880	4948	5016	5065	5160	5202	5319	5496	5556	5762	5836	5963	6121
	6193	6269	6282	6369	6402	6520	6567	6573	6706	6742	6820	6872	7050	7122	7169
	7281	7301	7448	7472	7652	7680	7693	7806	7810	7818	7822	7826	7862	7880	7973
	8107	8207	8300	8375	8498	8612	8701	8760	8839	8894	9135	9164	9430	9642	9722
	9981	9882	10136	10598	10599										
BISB	10491														
BIT	1528	1604	1671	1693	1955	1971	7582	7995	8129	8229	8239	9402	8404	8518	8520
	9281	9286	9291	9303	9366	9371	9376	9399	9794	9808	9816	9823	10283	10290	10299
BITB	9951														
BLOS	8890	10177													
BLT	9870	9886	9952	10046	10126	10235	10604								
BMI	9877														
BNE	1437	1460	1480	1520	1522	1598	1605	1672	1685	1726	1780	1821	1824	1827	1836
	1957	2010	2055	6932	7372	7577	7581	7583	7614	7622	7779	7815	8025	8152	8403
	8418	8519	8561	8727	8875	9041	9069	9211	9282	9284	9287	9289	9367	9369	9372
	9374	9396	9434	9466	9520	9619	9695	9724	9795	9824	9875	9932	9940	9948	9962
	9969	10016	10022	10028	10033	10041	10053	10063	10069	10097	10101	10107	10114	10121	10164
	10181	10187	10291	10328	10492	10504	10523	10594	10679						
BPL	9861	9891	9926	9966	10037	10065	10091	10296	10592						
BR	1422	1425	1462	1481	1486	1490	1495	1499	1503	1508	1512	1524	1565	1599	1608
	1612	1674	1700	1704	1708	1712	1755	1761	1767	1770	1841	1845	1849	1866	1875
	1881	1887	1914	1917	1920	1923	1926	1960	1964	1968	1974	2015	2059	2241	2290
	2449	3092	3751	4592	4641	5360	5404	5453	5578	5646	5717	5825	6221	7573	7584
	7612	7620	7631	8017	8154	8563	8636	8877	8886	8907	8935	8941	8966	9165	9294
	9299	9308	9316	9381	9387	9405	9431	9556	9562	9569	9573	9577	9584	9588	9708
	9714	9726	9732	9738	9742	9748	9754	9758	9766	9797	9803	9806	9819	9822	9872
	9889	9928	9945	9955	9964	9971	10031	10110	10137	10139	10183	10244	10257	10329	10335
	10343	10351	10359	10367	10375	10383	10391	10399	10407	10415	10423	10431	10439	10447	10455
	10463	10471	10479	10497	10506	10526	10533	10570	10585	10606	10671	10695			
CLR	1421	1424	1426	1427	1435	1448	1449	1643	1679	1728	1729	1818	1860	7573	7600
	7601	7607	7609	7610	7611	7629	7632	7633	7635	7636	7638	7772	7773	7787	7789
	8278	8279	8280	8421	8422	8423	8599	8600	8735	8742	8743	8914	8915	8946	8947
	8988	8989	9147	9237	9262	9263	9553	9821	9835	9864	9867	9997	10034	10080	10089
	10089	10153	10154	10228	10229	10485	10490	10583	10677						

K10

CLRB	1698	8019	8021	8156	8158	4820	9893	9944	9970	10189	10253					
CMP	1436	1459	1521	1601	1736	1738	1742	1744	1747	1749	1835	1862	1864	1897	1899	
	1902	1904	1907	1929	2018	2019	2062	2143	2152	2742	2751	3083	3097	4127	4136	
	4330	4339	4479	4488	7303	7312	7474	7483	7576	7602	7613	7621	8895	8887	8905	
	8908	8964	9173	9181	9310	9320	9360	9507	9514	9517	9682	9699	9692	9804	9828	
	9895	10015	10021	10023	10027	10032	10040	10045	10047	10052	10062	10058	10096	10106	10113	
	10125	10127	10163	10176	10327	10503	10505									
CMPB	1823	8013	8150	9810	9814	9937	9939	9947	9968	9972	10076	10100	10120	10180	10186	
	10232	10234														
DEC	1597	1684	1725	1779	1784	1852	2009	2054	6931	7371	7580	7778	7814	8024	8161	
	8417	8560	8726	8960	8992	9040	9068	9210	9283	9288	9368	9373	9395	9433	9465	
	9519	9618	9694	9723	10160	10498										
DECB	9951	9954	10591	10602												
FMT	65															
HALT	178	1848	2096	2325	2480	2604	2708	2955	3001	3063	3321	3430	3600	3702	3899	
	4042	4095	4297	4446	4676	4820	4932	5005	5149	5308	5485	5751	5352	6110	6182	
	6358	6503	6695	6726	6809	7039	7111	7270	7436	7650	7784	7962	8096	8198	8289	
	8264	8487	8593	8690	8823	9761	9771	9927	10297	10670	10694					
INC	1683	1776	1778	8012	8149	8723	8724	8725	8876	8952	8990	9690	9827	9871	10044	
	10051	10135	10162	10286	10597	10605	10676									
INCB	8020	8022	8157	8159	9832	9974	10280									
IOT	66															
JMP	196	198	200	1482	1619	1691	1695	1716	6933	6934	7373	7374	7568	7585	7737	
	8135	8565	8897	8916	8957	8968	9010	10020	10103	10330						
JSR	1478	1669	1814	1954	1996	2041	2084	2090	2094	2104	2161	2173	2219	2233	2245	
	2258	2274	2294	2315	2322	2351	2356	2361	2368	2375	2382	2407	2414	2419	2434	
	2453	2471	2477	2507	2512	2517	2524	2531	2538	2559	2596	2601	2629	2636	2646	
	2654	2664	2672	2680	2698	2705	2762	2767	2772	2780	2810	2815	2833	2852	2901	
	2909	2917	2922	2927	2932	2941	2960	2992	2998	3009	3029	3034	3052	3060	3066	
	3147	3155	3163	3168	3173	3178	3183	3192	3212	3265	3272	3284	3292	3311	3318	
	3349	3354	3359	3367	3390	3395	3403	3420	3427	3459	3467	3476	3483	3494	3514	
	3543	3550	3562	3570	3590	3597	3628	3633	3638	3645	3667	3675	3692	3699	3733	
	3738	3743	3755	3760	3765	3775	3784	3793	3800	3811	3832	3839	3849	3855	3863	
	3870	3889	3896	3927	3932	3937	3944	3953	3958	3972	3995	4033	4039	4070	4077	
	4084	4092	4147	4154	4167	4224	4231	4241	4253	4261	4269	4287	4294	4350	4355	
	4360	4369	4390	4405	4410	4418	4436	4443	4449	4504	4509	4518	4539	4569	4585	
	4526	4606	4625	4644	4666	4673	4718	4727	4746	4778	4785	4793	4810	4817	4862	
	4822	4887	4922	4929	4942	4963	4970	4978	4996	5003	5047	5058	5072	5107	5114	
	5122	5139	5146	5176	5186	5195	5209	5260	5267	5277	5288	5305	5323	5353	5365	
	5376	5397	5409	5419	5437	5456	5475	5482	5512	5517	5522	5531	5540	5545	5569	
	5574	5581	5586	5598	5626	5629	5651	5663	5671	5679	5686	5701	5720	5741	5748	
	5778	5783	5788	5797	5808	5821	5828	5844	5865	5882	5895	5903	5911	5918	5925	
	5943	5949	5979	5984	5989	5996	6027	6064	6072	6082	6090	6107	6136	6146	6154	
	6172	6179	6212	6217	6227	6232	6239	6248	6255	6264	6273	6289	6309	6315	6323	
	6331	6348	6355	6385	6390	6395	6409	6431	6459	6466	6474	6492	6499	6506	6527	
	6546	6555	6560	6579	6599	6641	6653	6658	6668	6675	6692	6723	6729	6760	6766	
	6774	6781	6799	6806	6844	6851	6858	6863	6876	6883	6896	6896	6920	6963	6975	
	6983	6993	7001	7011	7019	7036	7069	7075	7083	7101	7108	7146	7153	7160	7173	
	7180	7185	7190	7198	7222	7229	7235	7243	7260	7257	7323	7329	7334	7342	7359	
	7389	7396	7403	7410	7426	7433	7494	7499	7504	7509	7517	7534	7561	7596	7640	
	7647	7664	7671	7687	7698	7708	7714	7721	7759	7781	7792	7832	7841	7850	7955	
	7866	7885	7901	7920	7931	7942	7959	7997	8007	8054	8065	8076	8093	8131	8144	
	8171	8179	8195	8231	8241	8272	8286	8315	8324	8335	8344	8361	8400	8426	8437	
	8447	8458	8467	8484	8516	8535	8596	8590	8642	8650	8661	8670	8697	8746	8776	

MOV	8785	8794	8803	8820	8859	9005	9502	9512	9698	9717	9946	9953	9960	10018	10079
	10124	10292													
	1420	1423	1434	1438	1440	1441	1442	1443	1444	1445	1446	1447	1451	1452	1455
	1456	1457	1458	1463	1465	1466	1467	1471	1472	1473	1474	1475	1516	1530	1557
	1558	1570	1571	1575	1585	1587	1589	1590	1592	1594	1595	1596	1600	1602	1616
	1621	1623	1633	1634	1636	1637	1642	1644	1663	1666	1677	1678	1680	1681	1692
	1722	1723	1724	1727	1730	1731	1735	1758	1764	1772	1774	1782	1793	1791	1808
	1809	1811	1812	1822	1832	1851	1854	1861	1867	1878	1884	1890	1936	1937	1949
	1992	1994	2001	2002	2003	2005	2006	2008	2013	2025	2037	2039	2046	2047	2048
	2050	2051	2053	2057	2065	2085	2087	2088	2099	2111	2119	2122	2126	2127	2140
	2142	2146	2147	2150	2151	2155	2156	2216	2217	2326	2333	2336	2481	2488	2491
	2593	2594	2606	2614	2617	2710	2718	2721	2724	2725	2739	2741	2745	2746	2749
	2750	2754	2755	2843	2845	2846	2851	2857	2860	2872	2876	2878	2885	2989	2990
	3002	3013	3016	3054	3071	3074	3078	3081	3085	3086	3087	3088	3099	3100	3101
	3102	3107	3109	3120	3121	3122	3125	3131	3252	3263	3323	3331	3334	3432	3440
	3443	3540	3541	3602	3610	3613	3704	3712	3715	3901	3909	3912	4030	4031	4044
	4052	4055	4058	4059	4097	4105	4108	4110	4111	4124	4126	4130	4131	4134	4135
	4139	4140	4221	4222	4299	4307	4310	4313	4314	4327	4329	4333	4334	4337	4338
	4342	4343	4448	4456	4459	4462	4463	4476	4478	4482	4483	4496	4487	4491	4492
	4566	4567	4678	4686	4699	4709	4711	4713	4715	4737	4775	4776	4822	4830	4833
	4852	4855	4857	4859	4870	4919	4920	4934	4947	4950	5008	5015	5018	5038	5040
	5042	5044	5055	5104	5105	5151	5159	5162	5184	5257	5258	5310	5318	5321	5487
	5495	5498	5594	5753	5761	5764	5839	5954	5962	5965	6061	6062	6112	6120	5123
	6184	6192	6195	6279	6360	6368	6371	6399	6456	6457	6511	6519	6522	6570	6638
	6529	6645	6646	6697	6705	6708	6732	6741	6744	6811	6819	6822	6826	6827	6829
	6906	6907	6908	6910	6960	6961	6968	6969	7041	7049	7052	7113	7121	7124	7128
	7129	7131	7208	7209	7210	7212	7272	7280	7293	7286	7287	7300	7302	7306	7307
	7310	7311	7315	7316	7386	7387	7439	7447	7450	7453	7454	7456	7471	7473	7477
	7478	7481	7482	7486	7487	7558	7559	7574	7575	7586	7591	7605	7615	7616	7617
	7618	7619	7623	7624	7625	7626	7627	7628	7630	7634	7637	7651	7653	7756	7757
	7765	7766	7767	7769	7771	7774	7777	7790	7795	7798	7804	7805	7811	7875	7976
	7877	7917	7919	7924	7925	7926	7964	7972	7975	7979	7979	7994	8011	8015	9016
	8051	8052	8058	8059	8060	8098	8106	8109	8112	8113	8128	8148	8152	8153	8199
	8206	8209	8212	8213	8228	8238	8269	8270	8282	8291	8299	8302	8366	8374	8377
	8432	8442	8489	8497	8500	8583	8584	8598	8603	8611	8614	8635	8638	8692	8700
	8703	8734	8741	8751	8759	8762	8825	8838	8841	8892	8893	8895	8896	8913	8932
	8933	8938	8944	8954	8959	8962	8963	8967	8995	8999	9002	9032	9033	9034	9035
	9039	9042	9043	9061	9062	9063	9064	9065	9066	9067	9070	9071	9072	9089	9090
	9091	9092	9094	9095	9096	9108	9109	9110	9126	9127	9136	9140	9141	9142	9143
	9145	9146	9161	9163	9167	9169	9171	9176	9183	9203	9204	9205	9206	9207	9208
	9209	9212	9213	9214	9238	9239	9240	9264	9265	9266	9267	9268	9269	9270	9272
	9273	9274	9275	9276	9280	9285	9290	9302	9326	9327	9328	9329	9352	9353	9355
	9356	9358	9359	9361	9365	9370	9375	9394	9401	9408	9409	9423	9424	9425	9426
	9435	9436	9453	9454	9455	9456	9457	9458	9459	9460	9463	9467	9468	9469	9470
	9491	9492	9493	9494	9495	9496	9497	9498	9499	9500	9504	9509	9510	9511	9515
	9523	9524	9525	9526	9527	9554	9559	9565	9582	9592	9593	9611	9612	9613	9614
	9615	9616	9617	9620	9621	9622	9637	9638	9639	9640	9641	9643	9644	9646	9669
	9670	9671	9672	9673	9674	9675	9676	9677	9678	9679	9681	9685	9686	9690	9697
	9698	9699	9700	9701	9711	9719	9720	9729	9736	9745	9751	9770	9799	9900	9902
	9805	9818	9830	9831	9833	9834	9837	9838	9854	9855	9856	9857	9858	9859	9860
	9865	9868	9888	9894	9895	9896	9897	9898	9900	9901	9929	9930	9934	9949	9998
	9999	10000	10001	10003	10042	10054	10085	10102	10117	10122	10151	10152	10155	10165	10174
	10175	10190	10191	10192	10193	10220	10221	10222	10223	10224	10226	10227	10246	10247	10249
	10249	10250	10277	10282	10287	10301	10304	10325	10338	10346	10354	10362	10370	10379	10396

M10

	10394	10402	10410	10418	10426	10434	10442	10450	10458	10466	10474	10482	10489	10494	10507
	10508	10513	10518	10519	10521	10524	10528	10535	10536	10566	10574	10575	10576	10582	10589
	10607	10608	10609	10610	10611	10626	10627	10631	10659	10660	10661	10662	10663	10664	10665
	10666	10667	10668	10669	10675	10676	10680	10681	10682	10683	10684	10685	10686	10687	10688
	10691														
MOV8	1450	1525	1686	1873	9503	9836	9863	9866	9880	9883	9892	9931	9959	9967	10013
	10039	10050	10066	10081	10093	10161	10179	10184	10230	10299	10567	10568	10571	10572	10573
	10577	10580	10581	10600	10629										
NEG	9862	10578													
YOP	9006	9007	9008												
RESET	1431	1668	2083	9004											
RCR	10237	10239	10241	10594	10586	10587	10588	10590							
RTI	1732														
	1464	9241	9330	9410	9594	9839	9902	9936	10043	10055	10123	10156	10156	10194	10251
	10306	10612	10693												
RTS	2183	2392	2548	2568	2790	2950	3969	3202	3221	3376	3503	3522	3654	3820	3992
	4003	4178	4379	4399	4528	4548	4756	4897	5082	5219	5608	5854	5974	6006	6036
	6298	6419	6440	6589	6608	6911	6929	7213	7352	7368	7527	7543	7731	7895	8545
	8868	9044	9073	9097	9112	9128	9137	9148	9187	9190	9215	9437	9471	9528	9623
	9647	9702	9976	10004	10537	10632									
SUB	9093	9162	9168	9271	9309	9357	9427	9687	9869	10298					
TRAP	10634	10643	10644	10645	10646	10648	10650	10651	10652	10653	10654				
TST	1479	1519	1560	1787	1820	1825	1833	1934	1935	2227	2269	2429	3729	4579	4618
	5247	5391	5431	5563	5633	5696	5816	6208	7566	8630	8874	8990	8900	9948	9428
	9901	9825	9874	9884	9933	9941	9963	10002	10019	10030	10035	10078	10115	10130	10159
TSTB	10245	10252	10295	10302	10530	10595	10628								
.ASCII	1638	1689	9812	9876	9890	9925	9965	10036	10064	10090	10522				
	301	302	10826	10844	10860	10877	10912	10930	10948	10978	10988	11013	11022	11050	11058
	11078	11087	11111	11119	11136	11144	11161	11174	11193	11206	11223	11232	11244	11247	11262
	11274	11291	11295	11306	11314	11330	11345	11355	11376	11386	11403	11413	11432	11443	11471
	11484	11500	11510	11516	11532	11543	11556	11565	11570	11589	11601	11611	11654	11653	11683
	11692	11695	11708	11717	11724	11740	11746	11750	11765	11771	11777	11794	11802	11810	11823
	11830	11842	11849	11863	11869	11871	11880	11896	11916	11934	11949	11967	11987	12010	12025
.ASCIIZ	12042	12059	12069	12077											
	300	303	1488	1492	1497	1501	1505	1510	1514	1567	1610	1614	1676	1702	1706
	1710	1714	1757	1763	1769	1843	1847	1877	1893	1889	1916	1922	1928	1962	1966
	1970	1976	8937	8943	9013	9558	9564	9571	9575	9579	9596	9590	9710	9716	9729
	9734	9740	9744	9750	9756	9760	9768	10198	10199	10200	10201	10203	10337	10345	10353
	10351	10369	10377	10385	10393	10401	10409	10417	10425	10433	10441	10449	10457	10465	10473
	10481	10538	10697	10711	10715	10725	10734	10747	10754	10763	10772	10790	10792	10804	10918
	10834	10850	10867	10883	10896	10904	10918	10936	10956	10968	10995	11004	11029	11039	11065
	11074	11094	11103	11126	11151	11181	11213	11256	11292	11321	11341	11365	11393	11420	11450
	11461	11491	11523	11554	11577	11623	11637	11646	11670	11679	11704	11731	11755	11794	11814
	11834	11857	11889	11906	11925	11942	11955	11960	11977	11998	12017	12033	12051	12064	12073
	12084	12091	12103												
.BLKB	9985	10197													
.BLKW	1345	1346	1363	1379	9907										
.BYTE	256	257	262	263	271	272	280	281	282	283	1371	2276	2277	2296	2297
	2436	2437	2455	2456	2682	2683	2817	2818	3036	3037	3068	3069	3294	3295	3405
	3406	3572	3573	3677	3678	3872	3873	4079	4080	4271	4272	4420	4421	4627	4628
	4646	4647	4795	4796	4980	4981	5124	5125	5290	5291	5439	5440	5458	5459	5703
	5704	5722	5723	5927	5928	6092	6093	6156	6157	6333	6334	6494	6495	6677	6679
	6783	6784	7021	7022	7085	7086	7245	7246	7412	7413	7944	7945	8078	9079	9181
	9182	8346	9347	8469	8470	8672	8673	8905	8806	9012	10195	10196	10613	10614	10615

MEMO
DNR
NDC

10616	12159	12162	12165	12167	12169	12170	12174	12178	12181	12184	12186	12188	12190	12192
12196	9991													
10140														
12207	19	27	39	45	54	56	57	58	65	157	171	181	185	199
191	222	223	244	250	254	256	284	292	298	299	300	301	305	1006
1008	1053	1055	1344	1431	1438	1439	1442	1444	1446	1448	1449	1451	1452	1463
1488	1492	1497	1501	1505	1510	1514	1521	1527	1533	1555	1567	1579	1580	1583
1584	1585	1586	1610	1614	1627	1628	1631	1632	1633	1634	1650	1651	1661	1662
1663	1664	1676	1702	1706	1710	1714	1742	1752	1757	1763	1769	1795	1796	1806
1807	1809	1809	1810	1843	1847	1850	1871	1877	1883	1889	1897	1916	1922	1928
1930	1940	1941	1946	1947	1948	1962	1966	1970	1976	1982	1983	1989	1993	1995
2029	2030	2033	2034	2035	2072	2073	2078	2079	2080	2122	2126	2128	2142	2160
2161	2167	2203	2204	2214	2215	2216	2227	2231	2237	2238	2239	2240	2241	2249
2250	2251	2252	2253	2269	2272	2280	2284	2285	2286	2287	2300	2304	2305	2306
2307	2334	2336	2338	2339	2348	2361	2367	2368	2374	2418	2429	2432	2439	2443
2444	2445	2446	2458	2462	2463	2464	2465	2489	2491	2493	2494	2503	2517	2523
2524	2530	2576	2577	2591	2592	2593	2615	2617	2619	2620	2627	2640	2641	2642
2643	2658	2659	2660	2661	2696	2690	2691	2692	2693	2719	2721	2723	2724	2726
2741	2759	2820	2824	2825	2826	2827	2858	2860	2861	2862	2871	2898	2905	2906
2907	2908	2975	2976	2987	2988	2989	3014	3016	3018	3019	3026	3039	3043	3044
3045	3046	3079	3081	3082	3083	3119	3144	3151	3152	3153	3154	3230	3244	3251
3252	3250	3261	3262	3276	3277	3278	3279	3280	3298	3302	3303	3304	3305	3332
3334	3336	3337	3346	3408	3412	3413	3414	3415	3441	3443	3445	3446	3455	3458
3466	3467	3475	3476	3482	3491	3526	3527	3538	3539	3540	3554	3555	3556	3557
3558	3559	3576	3580	3581	3582	3583	3611	3613	3615	3616	3625	3680	3684	3685
3686	3687	3713	3715	3717	3718	3727	3743	3751	3765	3771	3775	3783	3792	3793
3799	3800	3808	3843	3844	3845	3846	3847	3848	3876	3880	3881	3882	3883	3910
3912	3914	3915	3924	3937	3943	3944	3952	3958	3964	3968	4020	4021	4028	4029
4030	4053	4055	4057	4058	4060	4068	4106	4108	4109	4110	4112	4126	4144	4147
4152	4154	4160	4169	4188	4189	4219	4220	4221	4235	4236	4237	4238	4257	4258
4259	4260	4275	4279	4280	4281	4282	4308	4310	4312	4313	4315	4329	4347	4348
4428	4429	4430	4431	4457	4459	4461	4462	4464	4478	4496	4557	4558	4564	4555
4566	4579	4583	4589	4590	4591	4592	4600	4601	4602	4603	4618	4622	4631	4635
4636	4637	4638	4650	4654	4655	4656	4657	4687	4689	4691	4692	4700	4718	4726
4727	4733	4737	4744	4766	4767	4773	4774	4775	4798	4802	4803	4804	4805	4831
4833	4835	4836	4844	4862	4870	4873	4879	4883	4907	4908	4917	4918	4919	4948
4950	4952	4953	4961	4984	4988	4989	4990	4991	5016	5018	5020	5021	5029	5047
5055	5058	5064	5068	5095	5096	5102	5103	5104	5127	5131	5132	5133	5134	5160
5162	5164	5165	5173	5176	5184	5186	5194	5195	5201	5205	5232	5233	5255	5256
5257	5271	5272	5273	5274	5287	5294	5298	5299	5300	5301	5319	5321	5323	5324
5332	5344	5347	5351	5357	5358	5359	5360	5369	5370	5371	5372	5391	5395	5401
5402	5403	5404	5413	5414	5415	5416	5431	5435	5443	5447	5449	5449	5450	5462
5466	5467	5468	5469	5496	5498	5500	5501	5509	5522	5530	5531	5539	5545	5553
5555	5559	5563	5567	5624	5633	5637	5643	5644	5645	5646	5655	5656	5657	5658
5696	5700	5707	5711	5712	5713	5714	5726	5730	5731	5732	5733	5762	5764	5766
5767	5775	5788	5796	5797	5805	5807	5816	5820	5839	5892	5899	5900	5901	5902
5931	5935	5936	5937	5938	5963	5965	5967	5968	5976	6049	6050	6059	6060	6061
6076	6077	6078	6079	6096	6100	6101	6102	6103	6121	6123	6125	6126	6134	6159
6163	6164	6165	6166	6193	6195	6197	6198	6206	6239	6247	6248	6254	6255	6263
6272	6273	6279	6336	6340	6341	6342	6343	6369	6371	6373	6374	6382	6446	6447
6454	6455	6456	6487	6491	6492	6493	6494	6520	6522	6524	6525	6533	6537	6545
6546	6554	6560	6566	6570	6619	6620	6636	6637	6638	6662	6663	6664	6665	6691

122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200

1997

1997

1997

	8112 8957 9932 10373 10642	8218 8959 9943 10395 10643	8223 8919 9943 10393 10644	8224 8624 9988 10401 10645	8249 8625 9004 10409 10646	9269 8713 9558 10417 10647	8307 8718 9564 10425 10648	8312 8719 9571 10433 10649	8313 8767 9575 10167 10441 10650	8389 8772 9579 10306 10449 10651	8394 8773 9586 10337 10457 10652	8395 8846 9590 10245 10465 10653	8505 8951 9710 10353 10473 10654	8510 8952 9716 10261 10481 10655	8511 8921 9729 10369 10634	
.PAGE	61 61	172 172	210 210	247 247	305 305	9101 9101	9129 9129	9149 9149	9191 9191	9530 9530	9624 9624	9763 9763	9841 9841	9908 9908	9978 9978	
.RADIX	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	2026 2026	
.REF	8510 8510	8511 8511	8624 8624	8625 8625	8718 8718	8719 8719	8772 8772	8773 8773	8851 8851	8852 8852	8851 8851	8852 8852	8851 8851	8852 8852	8851 8851	
.SBTTL	46 46	61 61	172 172	182 182	193 193	210 210	247 247	305 305	1239 1239	1412 1412	1413 1413	1414 1414	1417 1417	1432 1432	1519 1519	
.TITLE	1578 4187 8249 10541	1626 4556 8567 10618	1649 4765 8921 10634	1794 4906 8971 10655	1939 5094 8972 10655	1981 5231 8973 10655	1981 5231 8973 10655	2028 6048 8978 10655	2071 6445 9775 10655	2199 6618 9778 10655	2202 6936 9841 10655	2575 7377 9908 10655	2974 7547 9979 10655	3250 7746 10206 10655	3525 7903 10259 10655	4019 9030 10310 10655
.WORD	179 270 296 9011 12125	179 273 297 9529 12127	180 274 1338 9975 12129	190 275 1380 9982 12131	255 284 1381 9983 12134	258 286 1383 9984 12137	259 287 1387 10255 12139	260 288 1388 10258 12142	261 289 1402 10511 12145	261 290 1408 10516 12147	264 265 1409 10617 12149	265 291 1409 10617 12151	266 292 1410 10690 12151	267 293 1411 10592 12154	268 294 9993 12119	269 295 9996 12122

G11

MNDEC-11-DZJJA-A REF 5 6 FUNCT. CONT. TST-PT 2
DZJJA.P11 CROSS REFERENCE TABLE

MACY11 27(655) 30-MAR-76 18:59 PAGE 294

SEQ 0342

ERRORS DETECTED: 0

*DZJJA, DZJJA SOL CRF+DZJJA
RUN-TIME: 141 138 17 SECONDS
CORE USED: 29K

