

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DERPT-A-D
PRODUCT NAME: RP04 DISKLESS CONTROLLER TEST-PART II (STATIC 1B)
DATE CREATED: MAY 21, 1975
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: SUB MALLICK

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
7. RESTRICTIONS
8. MISCELLANEOUS
 - 8.1 EXECUTION TIME

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE RH70 AND DCL OF AN RP04 SUBSYSTEM. IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MDLI. IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MDLI OR BE APPROPRIATELY TERMINATED. IF THE DISK IS POWERED UP, IT IS REQUIRED TO GET THE DISK TO THE "HEADS UNLOADED" POSITION. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC IT CAN BE ASSERTED THAT, "THAT PART OF THE DCL THAT HANDLES DATA OR DATA ASSOCIATED LOGIC IS WORKING PROPERLY". THIS IMPLIES THAT, THAT PART OF THE LOGIC WHICH HANDLES MECHANICAL COMMANDS OR ITS ASSOCIATED LOGIC IS NOT TESTED IN THIS DIAGNOSTIC. ALL DATA COMMANDS USE THE MAINTENANCE REGISTER IN THE WRAPAROUND MODE.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04 DISK SYSTEM. THE RP04 DISK SYSTEM WILL CONSIST OF AN RH70 CONTROLLER, A DISK CONTROL LOGIC (DCL), THE CABLE FROM THE DCL CAN BE CONNECTED TO THE MDLI BUT IF NOT THAT CABLE MUST BE PROPERLY TERMINATED.

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

2.3 PRELIMINARY PROGRAMS

THIS PROGRAM ASSUMES THAT MAINDEC-11-DERPST- (LATEST REV) HAS BEEN RUN WITHOUT ERRORS

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

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

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

210 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS

STARTING ADDRESS THE CONSOLE TELETYPE WILL ASK FOR THE UNIT NUMBER TO BE TESTED. THEN ONLY THAT UNIT WILL BE TESTED FOR EACH PASS OF THE PROGRAM.

4.3 PROGRAM AND/OR OPERATOR ACTION

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

SWITCH 15 - HALT ON ERROR
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR THEN THE 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 6.

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: ALSO SEE SECTION 8.3

SWITCH 8 - LOOP ON TEST IN SWR <7:8>

THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 8 THRU 7 HAVE ONE MEANING AND WHEN RESET SWITCHES 8 THRU 7 HAVE ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH 8 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPRIATE POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 8 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 8 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 8 BEING LOW IS WITH SWITCH 6, WHICH IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH 8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST 120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,

COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: ALSO SEE SECTION 8.3

SWITCH 7 - STOP FURTHER COMPARES IF SW00 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 - ECC TEST-COMPARE END RESULTS ONLY IF SW00 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 ON ECC TESTS (TEST 120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE POSITION AND PATTERN REGISTERS AFTER EVERY CLOCK, COMPARES WILL BE DONE ONLY AT THE END OF ALL THE CLOCKS.

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES"

6.0 ERRORS

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

7.0 RESTRICTIONS

IF THERE IS A DRIVE CONNECTED THEN THE OPERATOR MUST HAVE THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B BUT NEVER LEAVE IT IN THE PROGRAMMABLE STATE. IF THERE IS NO DRIVE CONNECTED THEN THE CABLE NORMALLY GOING FROM THE DCL TO THE MDLI MUST BE PROPERLY TERMINATED.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE 1.75 MINUTES PER DRIVE. SUBSEQUENT PASSES WILL TAKE 7 MINUTE.

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. ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT THE PROGRAM GOES BACK TO CAN BE CHANGED. THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -

1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
2. LOOP ON ERROR SWITCH MUST BE SET
3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION

IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT COMES TO THE END OF THE TEST UNDER CONSIDERATION.

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

9.0 PROGRAM DESCRIPTION

THE FOLLOWING SECTIONS DESCRIBE EACH TEST AND SUBROUTINES IN DETAIL AND CAN ALSO BE USED AS AN INDEX TO THE LISTING. THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING WHERE THAT ITEM WILL BE FOUND.

DOCUMENT

MAINDEC-11-DERPTA-A

COPYRIGHT 1974
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

TABLE OF CONTENTS

13	OPERATIONAL SWITCH SETTINGS
27	BASIC DEFINITIONS
133	TRAP CATCHER
140	STARTING ADDRESS(ES)
151	MEMORY MANAGEMENT DEFINITIONS
190	COMMON TAGS
246	ERROR POINTER TABLE
908	REGISTER ADDRESSES
1074	REGISTER TEST
4595	END OF PASS ROUTINE
4637	SUBROUTINES
4717	SAVE REGISTERS ROUTINE
4745	FLOAT 1 AND 0
4782	CLEAR MEMORY ROUTINE
4815	LOCAL TRAPS
4832	CLEAD DISK ROUTINE
4845	CHECK DISK STATUS ROUTINE
4972	SAVE ROUTINE
4997	WRITE CHECK ROUTINE
5033	COMPARE ROUTINE
5123	CRC GENERATION ROUTINE
5436	JAM CURRENT CYLINDER ROUTINE

TABLE OF CONTENTS

5473	ECC GENERARION AND COMPARISON ROUTINE
5809	RH BASE ADDRESS CHANGE ROUTINE
5881	DISK SIMULATION
6850	SCOPE HANDLER ROUTINE
6924	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6992	TYPE ROUTINE
7039	TTY INPUT ROUTINE
7145	READ AN OCTAL NUMBER FROM THE TTY
7199	ERROR HANDLER ROUTINE
7245	ERROR MESSAGE TYPEOUT ROUTINE
7303	BINARY TO OCTAL (ASCII) AND TYPE
7381	TRAP DECODER
7396	TRAP TABLE
7420	POWER DOWN AND UP ROUTINES

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

PROGRAM BY SUB MALLICK

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

13

OPERATIONAL SWITCH SETTINGS

14

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SWR<7:0>
7	STOP FURTHER COMPARES IF SW00 IS LOW
6	ECC TEST-COMPARE END RESULTS ONLY IF SW00 IS LOW

27

BASIC DEFINITIONS

- 29 INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
- 40 GENERAL PURPOSE REGISTER DEFINITIONS
- 52 PRIORITY LEVEL DEFINITIONS
- 62 "SWITCH REGISTER" SWITCH DEFINITIONS
- 90 DATA BIT DEFINITIONS (BIT00 TO BIT15)
- 110 BASIC "CPU" TRAP VECTOR ADDRESSES

133 *****
TRAP CATCHER

136 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

140 *****
STARTING ADDRESS(ES)

146 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

151 *****
MEMORY MANAGEMENT DEFINITIONS

153 KT11 VECTOR ADDRESS

157 KT11 STATUS REGISTER ADDRESSES

164 KERNAL "I" PAGE DESCRIPTOR REGISTERS

175 KERNAL "I" PAGE ADDRESS REGISTERS

186 *****

188 *****

190 *****
COMMON TAGS

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

244 *****

246 *****
ERROR POINTER TABLE

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

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

263 *****
711 *****
716 *****
750 *****

908 *****
REGISTER ADDRESSES

1074 *****
REGISTER TEST

1158 *****
TEST 1 REFERENCE EACH REGISTER
 REFERENCE EACH REGISTER BY A MOVE INSTRUCTION

1202 *****
TEST 2 RHCS2-CONTROL AND STATUS 2

1205 THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
 OF THE NUMBER OF DRIVES PRESENT

1208 *****

1223 *****
TEST 3 PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT

1243 *****
TEST 4 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2

1353 *****
TEST 5 TEST SERIAL NUMBER AND DRIVE TYPE1
READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER
TYPE IT OUT AND PROCEED
TO LOOP HERE SET SWITCH 8 AND THIS TEST NO AND RESTART

1359 *****

1395 *****
TEST 6 CHECK MOL TO BE LOW

1398 MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE

1402 *****

1434 *****
TEST 7 PACK ACKNOWLEDGE COMMAND TEST

1437 THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
THEN ALL REGISTERS WILL BE CHECKED
RH CLEAR WILL BE GIVEN
THEN ALL REGISTERS WILL BE CHECKED

1442 *****

1501 *****
TEST 10 MAKE CURRENT CYLINDER = 0

1513 *****
TEST 11 CONTROL AND STATUS REGISTER 1 BITS 8 AND 9

1516 WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256
DATA IS THE CONTENTS OF THE TTY READER STATUS REGISTER
THIS WILL USE BITS A16 AND A17 WHEN THERE IS MORE THAN 28K OF MEMORY

1521 *****

1608 *****
TEST 12 DRIVE TIMING ERROR

1611 A READ HEADER AND DATA IS STARTED ON CYLINDER 0, SECTOR
0, TRACK 0, 260 WORDS. AFTER THE HEADER IS READ IN CORRECTLY
THEN NO SYNC BYTE (DATA SYNC) IS GIVEN.

1614 THEN NORMAL DIAGNOSTIC CLOCKS AND DIAGNOSTIC
 SECTOR CLOCKS ARE GIVEN FOR 24 BYTES,
 THEN 536 BYTES OF SECTOR CLOCKS ONLY ARE GIVEN,
 THIS IS TO TO BRING SECTOR PULSE UP
 THIS SHOULD SET DRIVE TIMING ERROR

1762
 TEST 13 DRIVE TIMING ERROR

1765 A WRITE DATA COMMAND IS STARTED ON CYLINDER 0, SECTOR
 0, TRACK 0, 256 WORDS. AFTER THE HEADER IS READ IN CORRECTLY
 THEN NO SYNC BYTE (DATA SYNC) IS GIVEN,
 THEN NORMAL DIAGNOSTIC CLOCKS AND DIAGNOSTIC
 SECTOR CLOCKS ARE GIVEN FOR 24 BYTES,
 THEN 536 BYTES OF SECTOR CLOCKS ONLY ARE GIVEN,
 THIS IS TO TO BRING SECTOR PULSE UP
 THIS SHOULD SET DRIVE TIMING ERROR

1900
 TEST 14 DRIVE TIMING ERROR

1911 A WRITE HEADER AND DATA COMMAND IS GIVEN
 TO CYLINDER 0, TRACK 0, 256. WORDS
 AFTER SECTOR IS FOUND (THE SECTOR FOUND FLOP IS HIGH)
 NO MORE DIAGNOSTIC CLOCKS ARE GIVEN,
 ONLY SECTOR CLOCKS ARE GIVEN TILL SECTOR PULSE IS HIGH
 THIS SHOULD SET DRIVE TIMING ERROR

2032
 TEST 15 SECTOR SELECTION
 THE SECTOR SELECTION LOGIC IS CHECKED HERE
 EACH SECTOR ON TRACK ZERO IS WRITTEN INTO
 DATA IS - 19 WORDS OF ZEROS - SYNC WORDS, 4 HEADER WORDS
 1 CRC WORD, 8 WORDS OF ZEROS, 1 SYNC WORD, 100 ZEROS
 (DATA), 1 SYNC WORD, 70 SECTOR NUMBER TO VARY

 THE WRITTEN DATA IS CHECKED IN MEMORY

2042
 2205
 TEST 16 WRITE ECC TEST 1

2200 THIS IS A WRITE ECC TEST
 WRITE CYLINDERS, FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
 OF ALL ZEROS.

2213

2353
TEST 17 READ ECC ENABLED 1A

2356 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
NO ERROR IS INSERTED
GOOD DATA USED IS 256 WORDS OF 0
COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

2363

2524
TEST 20 READ ECC ENABLED 1B

2527 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32
GOOD DATA USED IS 256 WORDS OF 0
COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD

2532 TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

2534

2701
TEST 21 READ ECC ENABLED 1C

2704 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 21 THRU 32
GOOD DATA USED IS 256 WORDS OF 0
COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

2711

2891
TEST 22 WRITE ECC TEST 2

2894 THIS IS A WRITE ECC TEST
WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
OF ALL ONES,

2899

3040 *****
TEST 23 READ ECC ENABLED 2A

3043 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
NO ERROR IS INSERTED
GOOD DATA USED IS 256 WORDS OF 177777
COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

3050 *****

3212 *****
TEST 24 READ ECC ENABLED 2B

3215 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32
GOOD DATA USED IS 256 WORDS OF 177777
COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

3222 *****

3389 *****
TEST 25 READ ECC ENABLED 2C

3392 THIS IS AN ECC READ DATA TEST
ERROR CORRECTION IS ENABLED
A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32 AND 21
GOOD DATA USED IS 256 WORDS OF 177777

3396 COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

3399 *****

3580 *****
TEST 26 WRITE ECC TEST 3

3583 THIS IS A WRITE ECC TEST
WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
OF ALL 52525.

3588 *****

3729 *****
TEST 27 READ ECC ENABLED 3A

3732 THIS IS AN ECC READ DATA TEST
 ERROR CORRECTION IS ENABLED
 NO ERROR IS INSERTED
 GOOD DATA USED IS 256 WORDS OF 52525
 COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

3739

3901

TEST 30 READ ECC ENABLED 3B

3904 THIS IS AN ECC READ DATA TEST
 ERROR CORRECTION IS ENABLED
 A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 4128
 THIS IS THE LAST BIT OF THE ECC
 GOOD DATA USED IS 256 WORDS OF 52525
 COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

3912

4085

TEST 31 READ ECC ENABLED 3C

4088 THIS IS AN ECC READ DATA TEST
 ERROR CORRECTION IS ENABLED
 A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 296 THRU 300
 THIS IS IN WORD NUMBER 19 AND 20
 GOOD DATA USED IS 256 WORDS OF 52525
 COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

4096

4275

TEST 32 READ ECC ENABLED 3D

4278 THIS IS AN ECC READ DATA TEST
 ERROR CORRECTION IS ENABLED
 A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32 AND 4096
 4096 IS THE LAST DATA BIT
 GOOD DATA USED IS 256 WORDS OF 52525
 COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
 TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

4286

4478

TEST 33 PROGRAM INTERRUPT

4473 PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
IN RHCS1 AT THE SAME TIME
THIS SHOULD INTERRUPT THROUGH LOCATION 254

4476 THE PROCESSOR PRIORITY IS SET TO 4
.....

4512
TEST 34 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

4515 PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
IE AND RDY IS SET, THIS SHOULD NOT INTERRUPT
.....

4548
.....
TEST 35 END OF DRIVE

4552 THIS IS THE END OF TEST FOR ONE DRIVE
IF THERE ARE MORE DRIVES THEN THE PROGRAM
JUMPS TO TEST 5 FOR NEXT DRIVE TEST
END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE

4557
4593

4595
END OF PASS ROUTINE
.....

4597 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

4637
SUBROUTINES
.....

4717 *****
SAVE REGISTERS ROUTINE

4745 *****
FLOAT 1 AND 0

4782 *****
CLEAR MEMORY ROUTINE

4815 *****
LOCAL TRAPS

4832 *****
CLEAD DISK ROUTINE

4845 *****
CHECK DISK STATUS ROUTINE

4972 *****
SAVE ROUTINE

4997 *****
WRITE CHECK ROUTINE

5033 *****
COMPARE ROUTINE

5123 *****
CRC GENERATION ROUTINE

5436 *****
JAM CURRENT CYLINDER ROUTINE

5473 *****
ECC GENERARION AND COMPARISON ROUTINE

5809 *****
RH BASE ADDRESS CHANGE ROUTINE

5810 THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
 ADDRESS FROM 176700 TO ANY TYPED VALUE

5881 *****
DISK SIMULATION

5882 *****

6028 *****

6050 *****

6057 *****

6184 *****

6251 *****

6319 *****

6392 *****

6424 *****

6541 *****

6685 *****

```

6790 .....
6800 .....
6847 .....

```

6850

```

.....
SCOPE HANDLER ROUTINE
.....

```

```

6852 THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS, IT WILL INCREMENT
AND LOAD THE TEST NUMBER(8TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
AND LOAD THE ERROR FLAG (8ERFLG) INTO DISPLAY<15:00>
THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
SW14=1 LOOP ON TEST
SW11=1 INHIBIT ITERATIONS
SW09=1 LOOP ON ERROR
SW08=1 LOOP ON TEST IN SWR<7:0>
CALL
      SCOPE          ;;SCOPE=IOT

```

```

6922 .....

```

6924

```

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

```

```

6926 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.
CALLI
      MOV      NUM,=(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
      TYPDS                      ;;GO TO THE ROUTINE

```

```

6990 .....

```

6992

.....
 TYPE ROUTINE

6994 ROUTINE TO TYPE ASCIZ MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE,
 THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.

CALL:
 1) USING A TRAP INSTRUCTION
 TYPE , MESADR ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 OR
 TYPE
 MESADR
 2) USING A JSR INSTRUCTION
 MOV PS, -(SP) ;; PUSH PROCESSOR STATUS WORD ON THE STACK
 JSR PC, \$TYPE ;; CALL TYPE ROUTINE
 MESADDR ;; FIRST ADDRESS OF MESSAGE

7037

7039

.....
 TTY INPUT ROUTINE

7040 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

7065 TK SERVICE ROUTINE
 THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT

7087

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

7107

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


```

7143  *****
*****
7145  READ AN OCTAL NUMBER FROM THE TTY
*****

7147  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 SHIOCT

7197  *****
*****
7199  ERROR HANDLER ROUTINE
*****

7201  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 SERRTYP 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=ENT AND N=ERROR ITEM NUMBER

7243  *****
*****
7245  ERROR MESSAGE TYPEOUT ROUTINE
*****

7247  THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
      ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
      AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

7300  *****
*****

```

7303

BINARY TO OCTAL (ASCII) AND TYPE

7305 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

```

CALLI
      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

```

STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
STYPOS OR STYPOC

```

CALLI
      MOV     NUM, -(SP)      ;;NUMBER TO BE TYPED
      TYPON   ;;CALL FOR TYPEOUT

```

STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER

```

CALLI
      MOV     NUM, -(SP)      ;;NUMBER TO BE TYPED
      TYPOC   ;;CALL FOR TYPEOUT

```

7379 *****

7381

TRAP DECODER

7383 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.

7396

TRAP TABLE

7398 THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
BY THE "TRAP" INSTRUCTION.

7418 *****

7420
POWER DOWN AND UP ROUTINES
.....

7460
.....

4745	OPERATIONAL SWITCH SETTINGS
4746	BASIC DEFINITIONS
4750	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
4757	MEMORY MANAGEMENT DEFINITIONS
4760	COMMON TAGS
(1)	ERROR POINTER TABLE
5410	REGISTER ADDRESSES
5577	REGISTER TEST
5619	T1 REFERENCE EACH REGISTER
5656	T2 RHCS2-CONTROL AND STATUS 2
5667	T3 PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
5679	T4 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
5761	T5 TEST SERIAL NUMBER AND DRIVE TYPEI
5790	T6 CHECK MOL TO BE LOW
5813	T7 PACK ACKNOWLEDGE COMMAND TEST
5869	T10 MAKE CURRENT CYLINDER = 0
5878	T11 CONTROL AND STATUS REGISTER 1 BITS 8 AND 9
5969	T12 DRIVE TIMING ERROR
6120	T13 DRIVE TIMING ERROR
6260	T14 DRIVE TIMING ERROR
6382	T15 SECTOR SELECTION
6543	T16 WRITE ECC TEST 1
6683	T17 READ ECC ENABLED 1A
6847	T20 READ ECC ENABLED 1B
7017	T21 READ ECC ENABLED 1C
7198	T22 WRITE ECC TEST 2
7339	T23 READ ECC ENABLED 2A
7504	T24 READ ECC ENABLED 2B
7674	T25 READ ECC ENABLED 2C
7856	T26 WRITE ECC TEST 3
7997	T27 READ ECC ENABLED 3A
8163	T30 READ ECC ENABLED 3B
8340	T31 READ ECC ENABLED 3C
8523	T32 READ ECC ENABLED 3D
8707	T33 PROGRAM INTERRUPT
8740	T34 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
8775	T35 END OF DRIVE
8808	END OF PASS ROUTINE
8812	SUBROUTINES
8871	SAVE REGISTERS ROUTINE
8894	FLOAT 1 AND 0
8931	CLEAR MEMORY ROUTINE
8960	LOCAL TRAPS
8977	CLEAD DISK ROUTINE
8990	CHECK DISK STATUS ROUTINE
9117	SAVE ROUTINE
9137	WRITE CHECK ROUTINE
9173	COMPARE ROUTINE
9253	CRC GENERATION ROUTINE
9547	JAM CURRENT CYLINDER ROUTINE
9582	ECC GENERARION AND COMPARISON ROUTINE
9894	RH BASE ADDRESS CHANGE ROUTINE
9942	DISK SIMULATION

10855	SCOPE HANDLER ROUTINE
10857	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10858	TYPE ROUTINE
10859	TTY INPUT ROUTINE
10864	READ AN OCTAL NUMBER FROM THE TTY
10866	ERROR HANDLER ROUTINE
10867	ERROR MESSAGE TIMEOUT ROUTINE
10869	BINARY TO OCTAL (ASCII) AND TYPE
10871	TRAP DECODER
(3)	TRAP TABLE
10879	POWER DOWN AND UP ROUTINES

(1)	000100	PR2=	100	;;PRIORITY LEVEL 2
(1)	000140	PR3=	140	;;PRIORITY LEVEL 3
(1)	000200	PR4=	200	;;PRIORITY LEVEL 4
(1)	000240	PR5=	240	;;PRIORITY LEVEL 5
(1)	000300	PR6=	300	;;PRIORITY LEVEL 6
(1)	000340	PR7=	340	;;PRIORITY LEVEL 7

;"SWITCH REGISTER" SWITCH DEFINITIONS

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

;"DATA BIT DEFINITIONS (BIT00 TO BIT15)

(1)	100000	BIT15=	100000
(1)	040000	BIT14=	40000
(1)	020000	BIT13=	20000
(1)	010000	BIT12=	10000
(1)	004000	BIT11=	4000
(1)	002000	BIT10=	2000
(1)	001000	BIT09=	1000
(1)	000400	BIT08=	400
(1)	000200	BIT07=	200
(1)	000100	BIT06=	100
(1)	000040	BIT05=	40
(1)	000020	BIT04=	20
(1)	000010	BIT03=	10
(1)	000004	BIT02=	4
(1)	000002	BIT01=	2
(1)	000001	BIT00=	1
(1)		.EQUIV	BIT09,BIT9
(1)		.EQUIV	BIT00,BIT0

```

(1)      .EQUIV BIT07,BIT7
(1)      .EQUIV BIT06,BIT6
(1)      .EQUIV BIT05,BIT5
(1)      .EQUIV BIT04,BIT4
(1)      .EQUIV BIT03,BIT3
(1)      .EQUIV BIT02,BIT2
(1)      .EQUIV BIT01,BIT1
(1)      .EQUIV BIT00,BIT0

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

4749
4750      .SBTTL TRAP CATCHER
(1)
(1)      .=0
(1)      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ",+2,HALT"
(1)      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

(1)      .SBTTL STARTING ADDRESS(ES)
(1)      .=200
(1)      000200 000137 004220      JMP      00BEGIN          ;)JUMP TO STARTING ADDRESS OF PROGRAM
4751      000210 000210
(1)      4752 000210 000137 004210      JMP      00BEGIN2       ;)JUMP SELECT TEST
4753      ;*STARTING ADDRESS 200 FOR NORMAL STARTS
4754      ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
4755      ;*
4756      ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
4757

(1)      .SBTTL MEMORY MANAGEMENT DEFINITIONS
(1)
(1)      ;*KT11 VECTOR ADDRESS
(1)
(1)      MHVEC= 250
(1)
(1)      ;*KT11 STATUS REGISTER ADDRESSES
(1)
(1)      SR0= 177572
(1)      SR1= 177574
(1)      SR2= 177576
(1)      SR3= 172516

```



```

4760 ;*****
(1)
(1) .SBTTL COMMON TAGS
(1)
(1) ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1) ;*USED IN THE PROGRAM.
(1)
(1) 000046 000046 .=46
(1) 000046 023352 SENDAD ;LOGICAL END OF PROGRAM
(1)
(1) 001100 .=1100
(1)
(1) 001100 SCNTAG; ;START OF COMMON TAGS
(1) 001100 000000 SPASS; .WORD 0 ;CONTAINS PASS COUNT
(1) 001102 000 STSTNM; .BYTE 0 ;CONTAINS THE TEST NUMBER
(1) 001103 000 SERFLG; .BYTE 0 ;CONTAINS ERROR FLAG
(1) 001104 000000 SICNT; .WORD 0 ;CONTAINS SUBTEST ITERATION COUNT
(1) 001106 000000 SLPADR; .WORD 0 ;CONTAINS SCOPE LOOP
(1) 001110 000000 SLPERR; .WORD 0 ;CONTAINS SCOPE RETURN FOR ERRORS
(1) 001112 000000 SERTTL; .WORD 0 ;CONTAINS TOTAL ERRORS DETECTED
(1) 001114 000 SITEMB; .BYTE 0 ;CONTAINS ITEM CONTROL BYTE
(1) 001115 001 SERMAX; .BYTE 1 ;CONTAINS MAX. ERRORS PER TEST
(1) 001116 000000 SERRPC; .WORD 0 ;CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001120 000000 SGDADR; .WORD 0 ;CONTAINS OF 'GOOD' DATA
(1) 001122 000000 SBDADR; .WORD 0 ;CONTAINS OF 'BAD' DATA
(1) 001124 000000 SGDDAT; .WORD 0 ;CONTAINS 'GOOD' DATA
(1) 001126 000000 SBDDAT; .WORD 0 ;CONTAINS 'BAD' DATA
(1) 001130 000000 000000 000000 .WORD 0,0,0 ;RESERVED--NOT TO BE USED
(1) 001136 177560 STKS; 177560 ;TTY KBD STATUS
(1) 001140 177562 STKB; 177562 ;TTY KBD BUFFER
(1) 001142 177564 STPS; 177564 ;TTY PRINTER STATUS REG.
(1) 001144 177566 STPB; 177566 ;TTY PRINTER BUFFER REG.
(1) 001146 000 SNULL; .BYTE 0 ;CONTAINS NULL CHARACTER FOR FILLS
(1) 001147 002 SFILLS; .BYTE 2 ;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001150 012 SFILLC; .BYTE 12 ;INSERT FILL CHARS, AFTER A "LINE FEED"
(1) 001151 000 STPFLG; .BYTE 0 ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(1) 001152 000000 SREGAD; .WORD 0 ;CONTAINS THE FROM
(1) ;WHICH (SREG0) WAS OBTAINED
(3) 001154 000000 SREG0; .WORD 0 ;CONTAINS ((SREGAD)+0)
(3) 001156 000000 SREG1; .WORD 0 ;CONTAINS ((SREGAD)+2)
(3) 001160 000000 SREG2; .WORD 0 ;CONTAINS ((SREGAD)+4)
(3) 001162 000000 SREG3; .WORD 0 ;CONTAINS ((SREGAD)+6)
(3) 001164 000000 SREG4; .WORD 0 ;CONTAINS ((SREGAD)+10)
(3) 001166 000000 SREG5; .WORD 0 ;CONTAINS ((SREGAD)+12)
(3) 001170 000000 STMP0; .WORD 0 ;USER DEFINED
(3) 001172 000000 STMP1; .WORD 0 ;USER DEFINED
(3) 001174 000000 STMP2; .WORD 0 ;USER DEFINED
(3) 001176 000000 STMP3; .WORD 0 ;USER DEFINED
(3) 001200 000000 STMP4; .WORD 0 ;USER DEFINED
(3) 001202 000000 STMP5; .WORD 0 ;USER DEFINED
(1) 001204 000000 STIMES; 0 ;MAX. NUMBER OF ITERATIONS
(1) 001206 000000 SESCAPE; 0 ;ESCAPE ON ERROR
(1) 001210 177607 000377 SBELL; .ASCIZ <207><377><377> ;CODE FOR BELL
(1) 001214 077 SQUES; .ASCII /?/ ;QUESTION MARK

```

(1) 001215 015
(1) 001216 000012

8CRLF: ,ASCII <15>
8LF: ,ASCIZ <12>

;;CARRIAGE RETURN
;;LINE FEED

4798					;PC OF JSR
4799					;TEST NO
4800					;WORD NO,
4801					;GOOD DATA
4802					;BAD DATA
4803					;CONTENTS OF RHCS1
4804					;CONTENTS OF RHDS1
4805					;CONTENTS OF RHER1
4806					
4807	001244	047402		DT32	;ERRPC,PCJSR,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
4808	001246	047706		DF32	10,0,0,1,0,0,0,0,0,
4809					
4810					
4811					
4812	001250	040213		EM2	;ERROR ON DATA COMMAND
4813					
4814	001252	045273		DH31	
4815					;PC
4816					;TEST NO
4817					;WORD NO,
4818					;GOOD DATA
4819					;BAD DATA
4820					;CONTENTS OF RHCS1
4821					;CONTENTS OF RHDS1
4822					;CONTENTS OF RHER1
4823	001254	047360		DT31	;ERRPC,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
4824	001256	047676		DF31	10,0,1,0,0,0,0,0,
4825					
4826					
4827					
4828					
4829	001260	000000		0	
4830	001262	000000		0	
4831	001264	047360		DT31	;ERRPC,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
4832	001266	047676		DF31	10,0,1,0,0,0,0,0,
4833					
4834					
4835					
4836	001270	040242		EM6	;ERROR ON WRITE HEADER AND DATA
4837					
4838	001272	045475		DH32	
4839					;PC
4840					;PC OF JSR
4841					;TEST NO
4842					;WORD NO,
4843					;GOOD DATA
4844					;BAD DATA
4845					;CONTENTS OF RHCS1
4846					;CONTENTS OF RHDS1
4847					;CONTENTS OF RHER1
4848	001274	047402		DT32	;ERRPC,PCJSR,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
4849	001276	047706		DF32	10,0,0,1,0,0,0,0,0,
4850					
4851					

4852					
4853					
4854	001300	040242		EM6	;ERROR ON WRITE HEADER AND DATA ;PC ;TEST NO ;WORD NO, ;GOOD DATA ;BAD DATA ;ERRPC,STSTNM,ERNORD,SGDDAT,SBDDAT ;0,0,1,0,0,
4855	001302	043057		DH2	
4856					
4857					
4858					
4859					
4860	001304	047102		DT3	
4861	001306	047565		DF3	
4862					
4863					
4864					
4865	001310	000000		0	;ERRPC,STSTNM,ERNORD,SGDDAT,SBDDAT ;0,0,1,0,0,
4866	001312	000000		0	
4867	001314	047102		DT3	
4868	001316	047565		DF3	
4869					
4870					
4871					
4872	001320	040301		EM11	;CONTROLLER OR DRIVE STATUS ;PC ;TEST NO ;FAILING REG, ADDR ;CONTENTS OF RNCB1 ;CONTENTS OF RNCB2 ;CONTENTS OF RNCB3 ;CONTENTS OF RNCB4 ;ERRPC,STSTNM,SBDDADR,CB1,CB2,DS1,ER1 ;0,0,0,0,0,0,
4873	001322	043202		DH11	
4874					
4875					
4876					
4877					
4878					
4879					
4880	001324	047116		DT11	
4881	001326	047572		DF11	
4882					
4883					
4884					
4885	001330	040301		EM11	;WRONG DATA FROM SILO ;PC ;REG,ADDR ;GOOD DATA ;RECEIVED DATA ;ERRPC,REGADR,SGDDAT,SBDDAT ;0,0,0,0,
4886					
4887	001332	042734		DH1	
4888					
4889					
4890					
4891	001334	047054		DT1	
4892	001336	047554		DF1	
4893					
4894					
4895					
4896	001340	000000		0	;ERRPC,STSTNM,REGADR,SGDDAT,SBDDAT ;0,0,0,0,0,
4897	001342	000000		0	
4898	001344	047054		DT1	
4899	001346	047554		DF1	
4900					
4901					
4902					
4903	001350	040334		EM14	;REGISTER FAILED ;PC ;FAILING REG, ADDR
4904	001352	043361		DH14	
4905					

4960			;ITEM21		
4961					
4962	001420	040513		EM21	; INTERRUPT FAILING
4963	001422	044263		DH21	; PC
4964					; TEST NO
4965					; CONTENTS OF RHC81
4966					; CONTENTS OF RHAS
4967					; CONTENTS OF RHDS1
4968	001424	047242		DT21	; SERRPC, TSTNM, CS1, AS, DS1
4969	001426	047635		DF21	; 0,0,0,0,0
4970					
4971					
4972			;ITEM22		
4973	001430	040535		EM22	; MISMATCH IN DRIVE PRESENT
4974					; LOOKING AT RHAS AND RHC82-NED(BIT012)
4975					; DRIVE PRESENT DO NOT AGREE
4976					; NOTE: ON DUAL PORT SYSTEM
4977					; DRIVE ON OTHER PORT WILL NOT GIVE NED
4978					; HENCE THERE WILL BE A MISMATCH
4979					; 177777-MEANS NOT PRESENT
4980	001432	044404		DH22	; PC
4981					; TEST NO
4982					; RHAS UNIT
4983					; RHC82 UNIT
4984					;
4985	001434	047256		DT22	; SERRPC, TSTNMS, SGDDAT, SBDDAT
4986	001436	047642		DF22	; 0,0,0,0
4987					
4988					
4989			;ITEM23		
4990	001440	000000		0	; MISMATCH IN DRIVE PRESENT
4991					; LOOKING AT RHAS AND RHC82-NED(BIT012)
4992					; DRIVE PRESENT DO NOT AGREE
4993					; 177777-MEANS NOT PRESENT
4994	001442	000000		0	; PC
4995					; TEST NO
4996					; RHAS UNIT
4997					; RHC82 UNIT
4998					;
4999	001444	047256		DT22	; SERRPC, TSTNMS, SGDDAT, SBDDAT
5000	001446	047642		DF22	; 0,0,0,0
5001					
5002					
5003					
5004			;ITEM 24		
5005	001450	041130		EM24	; LOOK AHEAD REGISTER AT THE
5006					; BEGINNING OF A SECTOR IS IN
5007					; ERROR
5008	001452	044503		DH24	; PC
5009					; RHDST
5010					; BAD RHLA
5011					; GOOD RHLA
5012					; SECTOR NO
5013					; SECTOR CLOCK

5014	001454	047270		DT24	;ERRPC,DST,0BDDAT,0TMP1,0TMP2,0TMP3
5015	001456	047646		DF24	;0,0,0,0,0
5016					
5017			;ITEM 25		
5018	001460	041223		EM25	;LOOK AHEAD REGISTER IS
5019					;IN ERROR
5020					
5021	001462	044503		DH24	;PC
5022					;RHDST
5023					;BAD RHLA
5024					;GOOD RHLA
5025					;SECTOR NO
5026					;SECTOR CLOCK
5027	001464	047270		DT24	;ERRPC,DST,0BDDAT,0TMP1,0TMP2,0TMP3
5028	001466	047646		DF24	;0,0,0,0,0
5029			;ITEM26		
5030	001470	040301		EM11	;CONTROLLER OR DRIVE STATUS
5031					
5032	001472	044641		DH26	;PC
5033					;PC OF JSR
5034					;FAILING REGISTER ADDRESS
5035					;CONTENTS OF RHCS1
5036					;CONTENTS OF RHCS2
5037					;CONTENTS OF RHDS1
5038					;CONTENTS OF RHER1
5039					
5040	001474	047306		DT26	;ERRPC,PCJSR,0BDADR,CS1,CS2,DS1,ER1
5041	001476	047654		DF26	;0,0,0,0,0,0,
5042					
5043					
5044					
5045			;ITEM27		
5046	001500	040130		EM1	;ERROR IN READING OR WRITING HARDWARE REGISTER
5047					
5048	001502	045023		DH27	;PC
5049					;PC OF JSR
5050					;TEST NUMBER
5051					;FAILING REGISTER
5052					;GOOD DATA
5053					;RECEIVED DATA
5054					
5055	001504	047326		DT27	;ERRPC,PCJSR,TSTNM,REGADR,0GDDAT,0BDDAT
5056	001506	047663		DF27	;0,0,0,0,0,0,
5057					
5058					
5059					
5060			;ITEM30		
5061	001510	041263		EM30	;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
5062	001512	045162		DH30	;PC
5063					;PC OF JSR
5064					;REGISTER ADDRESS
5065					;GOOD DATA
5066					;BAD DATA
5067					

5068	001514	047344	DT30	;ERRPC,PCJSR,REGADR,SGDDAT,SBDDAT
5069	001516	047671	DF30	;0,0,0,0,0
5070				
5071				
5072				
5073			;ITEM31	
5074	001520	041405	EM31	;ECC GENERATED IS INCORRECT
5075				;EVERY WORD IN THIS SECTOR IS GIVEN IN "DATA USED"
5076				
5077	001522	046131	DH34	;PC
5078				;TEST NUMBER
5079				;GOOD ECC1
5080				;GOOD EC2C
5081				;WRITTEN ECC1
5082				;WRITTEN ECC2
5083				;DATA USED
5084				
5085	001524	047450	DT34	;ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK
5086				
5087	001526	047727	DF34	;0,0,0,0,0,0,0
5088				
5089				
5090			;ITEM32	
5091	001530	041530	EM32	;ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ
5092				;ECC REGISTER OR RHER1 IS IN ERROR
5093				;ONLY LOWER 11 BITS OF PATTERN REGISTER
5094				;CAN BE READ
5095				;THIS SHUOLD MATCH LOWER 11 BITS OF ECC1
5096				
5097	001532	046304	DH35	;PC
5098				;TEST NUMBER
5099				;GOOD ECC1
5100				;GOOD ECC2
5101				;PATTERN REGISTER
5102				;RHER1
5103				
5104	001534	047470	DT35	;ERRPC,TSTNM,GECC1,GECC2,EC2,ER1
5105				
5106	001536	047736	DF35	;0,0,0,0,0,0
5107				
5108				
5109				
5110			;ITEM33	
5111	001540	042014	EM33	;HIGH COUNT BIT NOT HIGH AFTER 30859 CLOCKS
5112	001542	046500	DH36	;PC
5113				;PC OF JSR
5114				;TEST NUMBER
5115				;RHR
5116				;POSITION REG.
5117				;PATTERN REGISTER
5118				
5119	001544	047512	DT36	;ERRPC,PCJSR,TSTNM,HR,EC1,EC2
5120				
5121	001546	047746	DF36	;0,0,0,0,0,0

5122					
5123					
5124	001550	042066			
5125					
5126					
5127					
5128					
5129	001552	046500			
5130					
5131					
5132					
5133					
5134					
5135					
5136	001554	047512			
5137					
5138	001556	047746			
5139					
5140					
5141					
5142					
5143	001560	042161			
5144					
5145					
5146					
5147					
5148					
5149	001562	046636			
5150					
5151					
5152					
5153					
5154					
5155					
5156					
5157					
5158					
5159	001564	047530			
5160					
5161	001566	047754			
5162					
5163					
5164					
5165					
5166	001570	042460			
5167					
5168	001572	046304			
5169					
5170					
5171					
5172					
5173					
5174					
5175					

;ITEM34

EN34

;ZERO DETECT BIT NOT HIGH WHEN THE
 ;32 BIT ECC REGISTER HAS ITS 21 BITS
 ;OF ZEROS
 ;ERROR PRINTOUT WILL CONTINUE TILL
 ;ZERO DETECT BIT IS HIGH
 ;PC
 ;PC OF JSR
 ;TEST NUMBER
 ;RHR
 ;POSITION REG.
 ;PATTERN REGISTER

DH36

;ERRPC,PCJSR,TSTNN,MR,EC1,EC2

DF36

;0,0,0,0,0,0

;ITEM35

EN35

;POSITION REGISTER OR 11 BITS OF
 ;PATTERN REGISTER INCORRECT
 ;LOWER 11 BITS OF PATTERN REGISTER
 ;SHOULD MATCH LOWER 11 BITS OF GOOD ECC;
 ;DATA ENVELOPE AND N-CODE ZEROS ARE IN DECIMAL

DH37

;PC
 ;TEST NUMBER
 ;ECC POSITION
 ;GOOD POSITION
 ;GOOD ECC1
 ;GOOD ECC2
 ;ECC PATTERN
 ;DATA ENVELOPE
 ;N-CODE ZEROS

DT37

;ERRPC,TSTNN,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE

DF37

;0,0,0,0,0,0,0,0,0

;ITEM36

EN36

;ON A READ COMMAND WITH NON CORRECTABLE
 ;ERROR INSERTED DCK AND ECH SHOULD BE SET

DH35

;PC
 ;TEST NUMBER
 ;GOOD ECC1
 ;GOOD ECC2
 ;PATTERN REGISTER
 ;POSITION REGISTER
 ;RHR1

5176	001574	047470	DT35	;ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1
5177				
5178	001576	047736	DF35	;0,0,0,0,0,0,0
5179				
5180				
5181				
5182				
5183				
5184				
5185				
5186	001600	042646	;ITEM37 EM37	;ERROR ON DATA COMMAND ;WITH A16 A17 USED
5187				
5188				
5189	001602	045273	DH31	;PC ;TEST NO ;WORD NO. ;GOOD DATA ;BAD DATA ;CONTENTS OF RHCS1 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1
5190				
5191				
5192				
5193				
5194				
5195				
5196				
5197				
5198	001604	047360	DT31	;ERRPC,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
5199	001606	047676	DF31	;0,0,1,0,0,0,0,0,
5200				
5201				
5202				
5203				
5204	001610	000000	;ITEM40 0	
5205	001612	000000	0	
5206	001614	047360	DT31	;ERRPC,STSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1
5207	001616	047676	DF31	;0,0,1,0,0,0,0,0,
5208				
5209				

```

5211
5212 ;.....
5213 ;RH11 REGISTERS
5214
5215
5216
5217 ;.....
5218 ;WORD COUNT REGISTER (RHWC)
5219 ;EACH BIT IS CALLED BY BIT NUMBER
5220
5221
5222
5223 ;BUS ADDRESS REGISTER (RHBA)
5224 ;EACH BIT IS CALLED BY BIT NUMBER
5225
5226
5227
5228 ;CONTROL AND STATUS REGISTER 2 (RHCB2)
5229
5230 000001 US1= 1 ;UNIT SELECT (BIT 00)
5231 000002 US2= 2 ;UNIT SELECT (BIT 01)
5232 000004 US4= 4 ;UNIT SELECT (BIT 02)
5233 000010 BAI= 10 ;BUS ADDRESS INCREMENT INHIBIT (BIT 03)
5234 000020 PAT= 20 ;INVERT PARITY ON MASS BUS TO EVEN (BIT 04)
5235 000040 CLR= 40 ;CLEAR (BIT 05)
5236 000100 IR= 100 ;INPUT READY (BIT 06)
5237 000200 OR= 200 ;OUTPUT READY (BIT 07)
5238 000400 MPE= 400 ;MASS BUS PARITY ERROR (BIT 08)
5239 001000 MXP= 1000 ;MISSED TRANSFER ERROR (BIT 09)
5240 002000 PGE= 2000 ;PROGRAM ERROR (BIT 010)
5241 004000 NEM= 4000 ;NON EXISTANT MEMORY (BIT 011)
5242 010000 NED= 10000 ;NON EXISTANT DRIVE (BIT 012)
5243 020000 UPE= 20000 ;UNIBUS PARITY ERROR (BIT 013)
5244 040000 WCE= 40000 ;WRITE CHECK ERROR (BIT 014)
5245 100000 DLT= 100000 ;DATA LATE (BIT 015)
5246
5247 ;DATA BUFFER REGISTER (RHDB)
5248 ;EACH BIT IS CALLED BY BIT NUMBER
5249
5250
5251 ;.....
5252 ;RP04 REGISTERS
5253 ;.....
5254
5255
5256
5257 ;CONTROL AND STATUS 1 REGISTER, (000)
5258
5259 000001 GO= 1 ;GO (BIT 00)
5260 000100 IE= 100 ;INTERRUPT ENABLE (BIT 06)
5261 000200 RDY= 200 ;READY (BIT 07)
5262 000400 A16= 400 ;HIGH ORDER UNIBUS BITS (BIT 08)
5263 001000 A17= 1000 ;HIGH ORDER UNIBUS BITS (BIT 09)
5264 002000 PSEL= 2000 ;PORT SELECT (BIT 010)

```

5265	004000	DVA=	4000	;DEVICE AVAILABLE (BIT 011)
5266	020000	MCPE=	20000	;MASSBUS PARITY ERROR (BIT 013)
5267	040000	TRE=	40000	;TRANSFER ERROR (BIT 014)
5268	100000	SC=	100000	;SPECIAL CONDITION (BIT 015)
5269				
5270				
5271				
5272	000001	DFS=	1	;DRIVE FORWARD 5"/SEC. (BIT 00)
5273	000002	DFF20=	2	;DRIVE FORWARD 20"/SEC. (BIT 01)
5274	000004	DIGB=	4	;DRIVE TO INNER GAVRD BAND (BIT 02)
5275	000010	GRV=	10	;GO REVERSE (BIT 03)
5276	000020	DL64=	20	;DIFFERENCE LESS THAN 64 (BIT 04)
5277	000040	DE1=	40	;DIFFERENCE EQUALS 1 (BIT 05)
5278	000100	VV=	100	;VOLUME VALID (BIT 06)
5279	000200	DRY=	200	;DRIVE READY (BIT 07)
5280	000400	DPR=	400	;DRIVE PRESENT (BIT 08)
5281	001000	PROG=	1000	;PROGRAMABLE (BIT 09)
5282	002000	LST=	2000	;LAST SECTOR TRANSFERRED (BIT 010)
5283	004000	WRL=	4000	;WRITE LOCK (BIT 011)
5284	010000	MOL=	10000	;MEDIUM ON-LINE (BIT 012)
5285	020000	PIP=	20000	;POSITIONING OPERATION IN PROGRESS (BIT 013)
5286	040000	ERR=	40000	;COMPOSIT ERROR. (BIT 014)
5287	100000	ATA=	100000	;ATTENTION ACTIVE (BIT 015)
5288				
5289				
5290	000001			
5291	000002			
5292	000004			
5293	000010			
5294	000020			
5295	000040			
5296	000100			
5297	000200			
5298	000400			
5299	001000			
5300	002000			
5301	004000			
5302	010000			
5303	020000			
5304	040000			
5305	100000			
5306				
5307				
5308				
5309	000001			
5310	000002			
5311	000004			
5312	000010			
5313	000020			
5314	000040			
5315	000200			
5316	000400			
5317	001000			
5318				

;STATUS REGISTER (RHDS1) (001)

DFS= 1 ;DRIVE FORWARD 5"/SEC. (BIT 00)
 DFF20= 2 ;DRIVE FORWARD 20"/SEC. (BIT 01)
 DIGB= 4 ;DRIVE TO INNER GAVRD BAND (BIT 02)
 GRV= 10 ;GO REVERSE (BIT 03)
 DL64= 20 ;DIFFERENCE LESS THAN 64 (BIT 04)
 DE1= 40 ;DIFFERENCE EQUALS 1 (BIT 05)
 VV= 100 ;VOLUME VALID (BIT 06)
 DRY= 200 ;DRIVE READY (BIT 07)
 DPR= 400 ;DRIVE PRESENT (BIT 08)
 PROG= 1000 ;PROGRAMABLE (BIT 09)
 LST= 2000 ;LAST SECTOR TRANSFERRED (BIT 010)
 WRL= 4000 ;WRITE LOCK (BIT 011)
 MOL= 10000 ;MEDIUM ON-LINE (BIT 012)
 PIP= 20000 ;POSITIONING OPERATION IN PROGRESS (BIT 013)
 ERR= 40000 ;COMPOSIT ERROR. (BIT 014)
 ATA= 100000 ;ATTENTION ACTIVE (BIT 015)

;ERROR REGISTER 001 (RHER1) (002)

ILF= 1 ;ILLEGAL FUNCTION (BIT 00)
 ILR= 2 ;ILLEGAL REGISTER (BIT 01)
 RMR= 4 ;REGISTER MODIFICATION REFUSED (BIT 02)
 PAR= 10 ;PARITY ERROR (BIT 03)
 FER= 20 ;FORMAT ERROR (BIT 04)
 WCF= 40 ;WRITE CLOCK FAIL (BIT 05)
 ECH= 100 ;ECC HARD ERROR (BIT 06)
 HCE= 200 ;HEADER COMPARE ERROR (BIT 07)
 HCRC= 400 ;HEADER CRC ERROR (BIT 08)
 AOE= 1000 ;ADDRESS OVERFLOW ERROR (BIT 09)
 IAE= 2000 ;INVALID ADDRESS ERROR (BIT 010)
 WLE= 4000 ;WRITE LOCK ERROR (BIT 011)
 DTE= 10000 ;DRIVE TIMING ERROR (BIT 012)
 OPI= 20000 ;OPERATION INCOMPLETE (BIT 013)
 UNS= 40000 ;DRIVE UNSAFE (BIT 014)
 DCK= 100000 ;DATA CHECK ERROR (BIT 15)

;MAINTAINABILITY REGISTER (RHMR)(003)

DND= 1 ;DIAGINOSTIC MODE (BIT 00)
 NCLK= 2 ;MAINTAINABILITY CLOCK (BIT 01)
 MINX= 4 ;MAINTAINABILITY INDEX (BIT 02)
 MSTCK= 10 ;MAINTAINABILITY SECTOR CLOCK (BIT 03)
 MRD= 20 ;MAINTAINABILITY READ (BIT 04)
 MWR= 40 ;MAINTAINABILITY WRITE (BIT 05)
 DENVL= 200 ;DATA ENVELOPE (BIT 07)
 ZER= 400 ;ZERO DETECT (BIT 08)
 DTSY= 1000 ;MAINTAINABILITY SYNC DETECTED (BIT 09)

5319				
5320				
5321	000001	AT0=	1	;DEVICE 0 (BIT 00)
5322	000002	AT1=	2	;DEVICE 1 (BIT 01)
5323	000004	AT2=	4	;DEVICE 2 (BIT 02)
5324	000010	AT3=	10	;DEVICE 3 (BIT 03)
5325	000020	AT4=	20	;DEVICE 4 (BIT 04)
5326	000040	AT5=	40	;DEVICE 5 (BIT 05)
5327	000100	AT6=	100	;DEVICE 6 (BIT 06)
5328	000200	AT7=	200	;DEVICE 7 (BIT 07)
5329				
5330				
5331				;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (01)
5332				;EACH BIT IS CALLED BY BIT NUMBER
5333				;DRIVE TYPE REGISTER (RHDT) (006)
5334				;EACH BIT IS CALLED BY BIT NUMBER
5335				;LOOK-AHEAD REGISTER (RHLA) (007)
5336	000001	EXT1=	1	;EXTENSION 1 (BIT 00)
5337	000002	EXT2=	2	;EXTENSION 2 (BIT 01)
5338	000004	EXT4=	4	;EXTENSION 3 (BIT 02)
5339	000010	EXT10=	10	;EXTENSION 4 (BIT 03)
5340	000020	EXT20=	20	;EXTENSION 5 (BIT 04)
5341	000040	EXT40=	40	;EXTENSION 6 (BIT 05)
5342	000100	SC1=	100	;SECTOR COUNT FIELD 0 (BIT 06)
5343	000200	SC2=	200	;SECTOR COUNT FIELD 1 (BIT 07)
5344	000400	SC4=	400	;SECTOR COUNT FIELD 2 (BIT 08)
5345	001000	SC10=	1000	;SECTOR COUNT FIELD 3 (BIT 09)
5346	002000	SC20=	2000	;SECTOR COUNT FIELD 4 (BIT 010)
5347	004000	TRK1=	4000	;TRACK FIELD 1 (BIT 011)
5348	010000	TRK2=	10000	;TRACK FIELD 2 (BIT 012)
5349	020000	TRK4=	20000	;TRACK FIELD 3 (BIT 013)
5350	040000	TRK10=	40000	;TRACK FIELD 4 (BIT 014)
5351	100000	TRK20=	100000	;TRACK FIELD 5 (BIT 015)
5352				
5353				
5354				
5355	000001	WCUN=	1	;WRITE CURRENT UNSAFE (BIT 00)
5356	000002	CSFN=	2	;CURRENT SINK FAILURE (BIT 01)
5357	000004	WSUN=	4	;WRITE SELECT UNSAFE (BIT 02)
5358	000010	CSUN=	10	;CURRENT SWITCH UNSAFE (BIT 03)
5359	000020	MSEN=	20	;MOTOR SEQUENCE ERROR (BIT 04)
5360	000040	TDFN=	40	;TRANSITIONS DETECTOR FAILURE (BIT 05)
5361	000100	TUFN=	100	;TRANSITIONS UNSAFE (BIT 06)
5362	000200	FEN=	200	;FAILSAFE ENABLED (BIT 07)
5363	000400	WRUN=	400	;WRITE READY UNSAFE (BIT 08)
5364	001000	MHSN=	1000	;MULTIPLE HEAD SELECT (BIT 09)
5365	002000	NHSN=	2000	;NO HEAD SELECTION (BIT 010)
5366	004000	IXEN=	4000	;INDEX ERROR (BIT 011)
5367	010000	VU30=	10000	;30VOLT UNSAFE (BIT 012)
5368	020000	PLUN=	20000	;PLO UNSAFE (BIT 013)
5369	100000	ACUN=	100000	;ACUNSAFE (BIT 015)
5370				
5371				
5372				;OFFSET REGISTER (RHOF) (011)

5373	000001	OF25=	1	;OFFSET 25 MICRO INCHES (BIT 00)
5374	000002	OF50=	2	;OFFSET 50 MICRO INCHES (BIT 01)
5375	000004	OF100=	4	;OFFSET 100 MICRO INCHES (BIT 02)
5376	000010	OF200=	10	;OFFSET 200 MICRO INCHES (BIT 03)
5377	000020	OF400=	20	;OFFSET 400 MICRO INCHES (BIT 04)
5378	000040	OF800=	40	;OFFSET 800 MICRO INCHES (BIT 05)
5379				
5380	000200	OPREV=	200	;OFFSET NEGATIVE (REVERSE) (BIT 07)
5381	002000	HCI=	2000	;HEADER COMPARE INHIBIT (BIT 010)
5382	004000	ECI=	4000	;ERROR CORRECTION CODE INHIBIT (BIT 011)
5383	010000	FMT22=	10000	;FORMAT BIT (BIT 012)
5384				
5385				
5386				
5387				
5388				
5389				
5390				
5391				
5392				
5393	000001	PSU=	1	;PACK SPEED UNSAFE (BIT 00)
5394	000002	VUF=	2	;VELOCITY UNSAFE (BIT 01)
5395	000010	UNR=	10	;ANY UNSAFE EXCEPT READ/WRITE (BIT 03)
5396	000020	PRE=	20	;DISK PACK ROTATION ERROR (BIT 04)
5397	000040	ACL=	40	;AC LOW (BIT 05)
5398	000100	DCL=	100	;DC LOW (BIT 06)
5399	040000	SKI=	40000	;SEEK INCOMPLETE (BIT 014)
5400	100000	OCYL=	100000	;OFF CYLINDER (BIT 015)
5401				
5402				
5403				
5404				
5405				

;DESIRED CYLINDER ADDRESS (RHCA) (012)
 ;EACH BIT IS CALLED BY BIT NUMBER
 ;CURRENT CYLINDER ADDRESS (RHCC) (013)
 ;EACH BIT IS CALLED BY BIT NUMBER
 ;SERIAL NUMBER REGISTER (RHSN) (014)
 ;EACH IS CALLED BY BIT NUMBER
 ;ERROR REGISTER 003 (RHER3) (015)

;ECC POSITION REGISTER (RHEC1) (016)
 ;EACH BIT IS CALLED BY BIT NUMBER
 ;ECC PATTERN REGISTER (RHEC2) (017)
 ;EACH BIT IS CALLED BY BIT NUMBER

5407
 5408
 5409
 5410
 5411
 5412
 5413
 5414
 5415
 5416
 5417
 5418
 5419
 5420
 5421
 5422
 5423
 5424
 5425
 5426
 5427
 5428
 5429
 5430
 5431
 5432
 5433
 5434
 5435
 5436
 5437
 5438
 5439
 5440
 5441
 5442
 5443
 5444
 5445
 5446
 5447
 5448
 5449
 5450
 5451
 5452
 5453
 5454
 5455
 5456
 5457
 5458
 5459

001620 000254

001622 176722
 001624 176702
 001626 176704
 001630 176710
 001632 176700
 001634 176714
 001636 176706
 001640 176740
 001642 176732
 001644 176734
 001646 176742
 001650 176716
 001652 176724
 001654 176712
 001656 176726
 001660 176730
 001662 176744
 001664 176746
 001666 176720
 001670 176736

.SBTTL REGISTER ADDRESSES

;RP04 VECTOR ADDRESS

RPVEC: 254

;RP04 VECTOR ADDRESS

;RP04 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER
;NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
IF THE "CHANGE BASE ADDRESS" ROUTINE IS USED,
THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

RHDB: 176722
 RHWC: 176702
 RHBA: 176704
 RHCS2: 176710

;DATA BUFFER SEE NOTE ABOVE
;WORD COUNT SEE NOTE ABOVE
;BUS ADDRESS SEE NOTE ABOVE
;CONTROL AND STATUS 2 SEE NOTE ABOVE

;RP04 DISK I/O REGISTERS LOCATED IN THE RP04 DEVICE LOGIC
;NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
IF THE "CHANGE BASE ADDRESS ROUTINE IS USED,
THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

RHCS1: 176700
 RHER1: 176714
 RHDST: 176706
 RHER2: 176740
 RHOF: 176732
 RHCA: 176734
 RHER3: 176742
 RHAS: 176716
 RHR: 176724
 RHDS1: 176712
 RHDT: 176726
 RHRN: 176730
 RHEC1: 176744
 RHEC2: 176746
 RHLA: 176720
 RHCC: 176736

;CONTROL AND STATUS 1 SEE NOTE ABOVE
;ERROR 01 SEE NOTE ABOVE
;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE
;ERROR 02 SEE NOTE ABOVE
;OFFSET SEE NOTE ABOVE
;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE
;ERROR 03 SEE NOTE ABOVE
;ATTENTION SUMMARY SEE NOTE ABOVE
;MAINTAINABILITY SEE NOTE ABOVE
;DRIVE STATUS SEE NOTE ABOVE
;DRIVE TYPE SEE NOTE ABOVE
;SERIAL NUMBER SEE NOTE ABOVE
;ECC POSITION SEE NOTE ABOVE
;ECC PATTERN SEE NOTE ABOVE
;LOOK-AHEAD SEE NOTE ABOVE
;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE

```

5461
5462
5463
5464
5465
5466
5467 001672 000000
5468 001674 000000
5469 001676 000000
5470 001700 000000
5471
5472
5473 001702 000000
5474 001704 000000
5475 001706 000000
5476 001710 000000
5477 001712 000000
5478 001714 000000
5479 001716 000000
5480 001720 000000
5481 001722 000000
5482 001724 000000
5483 001726 000000
5484 001730 000000
5485 001732 000000
5486 001734 000000
5487 001736 000000
5488 001740 000000
5489
5490
5491
5492 001742 000010
5493 001762 000000
5494 001764 000000
5495
5496 001766 000000
5497
5498 001770 000000
5499 001772 000000
5500
5501
5502
5503 001774 000000
5504
5505
5506
5507 001776 000000
5508
5509
5510 002000 000000
5511
5512
5513
5514

```

;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTERS
 ;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: 0 ;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
NR: 0 ;MAINTAINABILITY
DS1: 0 ;DRIVE STATUS
DT: 0 ;DRIVE TYPE
SN: 0 ;SERIAL NUMBER
EC1: 0 ;ECC POSITION
EC2: 0 ;ECC PATTERN
LA: 0 ;LOOK-AHEAD
CC: 0 ;CURRENT CYLINDER ADDRESS

UNITS: .BLKW 0. ;THIS IS FILLED WITH -1
UNIT: .WORD 0 ;UNIT UNDER TEST
NOUNIT: .WORD 0 ;NUMBER OF UNITS PRESENT
;USED TO KEEP TRACK OF UNIT UNDER TEST
NUNIT: .WORD 0 ;USED TO DETERMIN IF THERE ARE MORE
;THAN ONE UNIT
SELECT: .WORD 0 ;ALL ONES INDICATE UNIT TO BE SELECTED
UNITSL: .WORD 0 ;UNIT NO. SELECTED

ERFLG: 0 ;ERROR FLAG

SAVDT: 0 ;SAVE DRIVE TYPE REGISTER
;FOR COMPARISON IN DRIVE CLEAR TEST
;AND RH INIT TEST
SAVSN: 0 ;SAVE SERIAL NUMBER REGISTER
;FOR COMPARISON IN DRIVE CLEAR TEST
;AND RH INIT TEST

```

5515				
5516	002002	000000	PCJSR: 0	;SAVE PC OF JSR WHICH GAVE THE ERROR
5517				
5518				
5519				
5520	002004	000000	ATTENT: 0	;ATTENTION BIT FOR PRESENT UNIT
5521	002006	000000	TOTALAT: 0	;TATAL ATTENTION BITS
5522				
5523				
5524	002010	000000	TMPILL: 0	;TEMPORARY ILLEGAL FUNCTION
5525				
5526				
5527	002012	000000	TSECC: 0	;FLAG TO SAY IF ECC TEST OR NOT
5528				;WHEN = 177777 IT IS AN ECC TEST
5529				;WHEN = 0 IT IS NOT AN ECC TEST
5530				
5531	002014	000000	TESDTE: 0	;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
5532				;WHEN = 177777 IT IS A DTE TEST
5533				;WHEN = 0 IT IS NOT A DTE TEST
5534				
5535				
5536	002016	000000	TAGDTE: 0	;TEMPORARY TAG USED IN DRIVE TIMING
5537				;ERROR TEST


```

5577          ,SBTTL REGISTER TEST
5578 004210 012737 177777 001770 BEGIN2: MOV 0-1,00SELECT ;SELECT UNIT
5579 004216 000402          BR START
5580 004220 005037 001770 BEGIN: CLR 00SELECT ;DO NOT SELECT UNIT
5581          ;NORMAL RUN
5582
5583          START:
5584 004224 012737 000340 177776 MOV 0340,00PS ;;LOCK OUT ALL INTERRUPTS
(1) 004232 012706 001100 MOV 00CNTAG,R6 ;;FIRST LOCATION TO BE CLEARED
(1) 004236 005026          CLR (R6)+ ;;CLEAR MEMORY LOCATION
(1) 004240 022706 001136 CMP 00TKS,R6 ;;DONE?
(1) 004244 001374          BNE ,-6 ;;LOOP BACK IF NO
(1) 004246 012706 001000 MOV 00STACK,SP ;;SETUP THE STACK POINTER
(1) 004252 012737 035460 000020 MOV 00SCOPE,00IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1) 004260 012737 000340 000022 MOV 0340,00IOTVEC+2 ;;LEVEL 7
(1) 004266 012737 037152 000030 MOV 00ERROR,00ENTVEC ;;ENT VECTOR FOR ERROR ROUTINE
(1) 004274 012737 000340 000032 MOV 0340,00ENTVEC+2 ;;LEVEL 7
(1) 004302 012737 037716 000034 MOV 00TRAP,00TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
(1) 004310 012737 000340 000036 MOV 0340,00TRAPVEC+2 ;;LEVEL 7
(1) 004316 012737 037764 000024 MOV 00PWRDN,00PWRVEC ;;POWER FAILURE VECTOR
(1) 004324 012737 000340 000026 MOV 0340,00PWRVEC+2 ;;LEVEL 7
(1) 004332 005067 174646          CLR 00TIMES ;;INITIALIZE NUMBER OF ITERATIONS
(1) 004336 005067 174644          CLR 00ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 004342 112767 000001 174545 MOVB 01,00ERMAX ;;ALLOW ONE ERROR PER TEST
(1) 004350 012767 004350 174530 MOV 0,,00LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 004356 012767 004356 174524 MOV 0,,00LPERR ;;SETUP THE ERROR LOOP ADDRESS
5588
5589
5590 004364 012767 000000 173404 MOV 00,PS ;;SET PROCESSOR STATUS TO 0
5591 004372 012777 035416 175220 MOV 00RPVECT,00RPVEC ;;THIS IS FOR UNTIMELY RPO4 INTERRUPTS
5592 004400 004737 036424          JSR PC,000TKINT ;;INITILIZE TK
5593 004404 005737 004176          TST 00FIRST ;;IS THIS FIRST TIME ROUND
5594 004410 001001          BNE 10 ;;BRANCH IF NOT
5595 004412 000402          BR 20
5596 004414 000137 005232          JMP 000SND1
5597 004420          10:
(1) 004420 104400 004426          20:
(1) 004424 000437          TYPE ,,+4 ;;TYPE ASCII STRING
(1)          BR 648 ;;GET OVER THE ASCII
(1)          ;,ASCII <15><12>/RPO4 DISKLESS CONTROLLER TEST-PART II (STATIC 15) - DERPT
(1) 004524          648:
5598 004524 104400 004532          TYPE ,,+4 ;;TYPE ASCII STRING
(1) 004530 000425          BR 658 ;;GET OVER THE ASCII
(1)          ;,ASCII <15><12>/MAKE SURE DCL IS LOCKED ON CORRECT PORT/
(1) 004604          658:
5599 004604 104400 004612          TYPE ,,+4 ;;TYPE ASCII STRING
(1) 004610 000425          BR 668 ;;GET OVER THE ASCII
(1)          ;,ASCII <15><12>/IF CHANGES ARE REQUIRED ON SWITCH THEN/
(1) 004664          668:
5600 004664 104400 004672          TYPE ,,+4 ;;TYPE ASCII STRING
(1) 004670 000430          BR 678 ;;GET OVER THE ASCII
(1)          ;,ASCII <15><12>/A CYCLE UP SEQUENCE IS REQUIRED FOR STROBING/
(1) 004752          678:
5601 004752 104400 004760          TYPE ,,+4 ;;TYPE ASCII STRING
(1) 004756 000414          BR 688 ;;GET OVER THE ASCII

```

```

(1)
(1) 005010
5602 005010 104400 005016
(1) 005014 000432
(1)
(1) 005102
5603 005102 104400 005110
(1) 005106 000427
(1)
(1) 005166
5604 005166 104400 005174
(1) 005172 000417
(1)
(1) 005232
5605 005232 012737 177777 004176
5606 005240 005737 001770
5607
(1) 005244 001435
(1)
5608 005246 104400 005254
(1) 005252 000423
(1)
(1) 005322
5609 005322 104416
5610 005324 042716 177770
5611 005330 011637 001762
5612 005334 012637 001772
5613
5614
5615
5619
(3)
(4)
(3)
(2) 005340 000004
(1) 005342 012767 000001 173634
5620 005350 012706 001000
5621
(1) 005354 012737 000001 004174
(1)
5622 005362 012737 037160 000030
5623
5624 005370 012737 005416 000004
5625 005376 012700 000024
5626 005402 012701 001622
5627 005406 013102
5628 005410 005300
5629 005412 001375
5630 005414 000471
5631 005416 012737 000006 000004
5632 005424 022626
5633 005426 016167 177776 173536
5634 005434 104015
5635 005436 032737 020000 177570

        ;;ASCIZ      <15><12>/THE PORT SELECT FLOP/
608:    TYPE      ,,+4          ;;TYPE ASCIZ STRING
        BR        698          ;;GET OVER THE ASCIZ
        ;;ASCIZ      <15><12>/ALL DCL UNDER TEST MUST BE LOCKED ON CORRECT PORT/
698:    TYPE      ,,+4          ;;TYPE ASCIZ STRING
        BR        708          ;;GET OVER THE ASCIZ
        ;;ASCIZ      <15><12>/ALL DCL NOT UNDER TEST MUST BE SWITCHED OFF/
708:    TYPE      ,,+4          ;;TYPE ASCIZ STRING
        BR        718          ;;GET OVER THE ASCIZ
        ;;ASCIZ      <15><12>/OR LOCKED ON THE OTHER PORT/
718:    SND1:    MOV      0-1,0#FIRST          ;NEXT TIME DO NOT GIVE HEADER
        TST      0#SELECT          ;WAS IT A 200 START
        BEQ      TST1          ;BRANCH IF STARTING FROM 200
        TYPE      ,,+4          ;;TYPE ASCIZ STRING
        BR        648          ;;GET OVER THE ASCIZ
        ;;ASCIZ      <15><12>/SELECT UNIT NUMBER TO BE TESTED ? /
648:    RDOCT
        BIC      0177770,(SP)          ;ONLY KEEP LAST 3 BITS
        MOV      (SP),0#UNIT          ;SAVE UNIT TO BE TESTED
        MOV      (SP)+,0#UNITSL       ;SAVE UNIT TO BE TESTED

;*****
;TEST 1            REFERENCE EACH REGISTER
;#                REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
;*****
TST1:   SCOPE
        MOV      0-1,0#TIMES          ;;DO 1 ITERATION
        MOV      0#STACK, SP          ;SET UP STACK POINTER
        MOV      0#TTNO,0#TSTNM      ;THIS SAVES TEST NUMBER
        MOV      0#REGSA1,0#ENTVEC;ERROR VECTOR SO THAT
        ;NO REGISTERS ARE SAVED
        MOV      028, 0#ERRVEC ;SET UP FOR BUS TIMEOUT
        MOV      024, R0          ;THERE ARE 24 REG TO TEST
        MOV      0#RHDB, R1          ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
        MOV      0(R1)+, R2          ;READ HARDWARE REG.
        DEC      R0          ;COUNT DOWN
        BNE     18          ;BRANCH IF 24 NOT DONE
        BR      38          ;BRANCH IF 24 DONE
        MOV      0#ERRVEC+2,0#ERRVEC ;RESTORE TRAP CATCHER
        CMP      (SP)+, (SP)+ ;CLEAN STACK
        MOV      -2(R1), 0#TMP1 ;STORE FAILING REG ADDR
        ERROR   15          ;REGISTER NON EXISTANT
        BIT     0#SW13,0#SWR        ;INHIBIT ERROR PRINTOUT ?
  
```

```
5636 005444 001053          BNE      48          ;BRANCH IF YES
5637 005446 104400 005454    TYPE     ,,+4       ;;TYPE ASCIZ STRING
(1) 005452 000431          BR       648       ;;GET OVER THE ASCIZ
(1)          ;;ASCIZ    <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM /
(1) 005536          648:
5638 005536 104400 005544    TYPE     ,,+4       ;;TYPE ASCIZ STRING
(1) 005542 000411          BR       658       ;;GET OVER THE ASCIZ
(1)          ;;ASCIZ    <15><12>/AND RESTART AT /
(1) 005566          658:
5639 005566 012746 030056    MOV      @BASECH,-(SP) ;GET READY TO TYPE STARTING ADDRESS
5640          ;OF "CHANGE OF BASE ADDRESS" ROUTINE
5641 005572 104402          TYP0C
5642 005574 000137 023246    JMP      @@EOP      ;GO TO END OF PROGRAM
5643 005600 012737 037152 000030 38:  MOV      @@ERROR,@@ENTVEC;RESTORE ERROR VECTOR
5644          ;SO THAT REGISTERS ARE SAVED
5645 005606 012737 000006 000004    MOV      @ERRVEC+2,@@ERRVEC ;RESTORE TRAP CATCHER
5646
5647
5648
5655
5656          ;*****
(3)          ;*TEST 2          RHCS2-CONTROL AND STATUS 2
(4)
(4)          ;*          THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
(4)          ;*          OF THE NUMBER OF DRIVES PRESENT
(4)
(3)          ;*****
(2) 005614 000004          TST2:  SCOPE
(1) 005616 012767 000001 173360    MOV      @1,@TIMES   ;;DO 1 ITERATION
5657 005624 012706 001000          MOV      @STACK,SP   ;RESET STACK
5658
5659          UN:
(1) 005630 012737 000002 004174    MOV      @TTNO,@@TSTNM ;THIS SAVES TEST NUMBER
(1)
5660 005636 013737 001630 005652    MOV      @@RHCS2,@@UN+2
5661 005644 004537 024236          JSR      R5,@@BITST  ;TEST BITS IN REGISTER
5662          ;ONLY THESE BITS ARE TEST READ/WRITE
5663          ;ADDRESS OF REG, BEING TESTED
5664          ;IN CORRECT DATA RECEIVED
5665          ;RETURN TO BLT) ROUTINE
5666          ;*****
5667          ;*TEST 3          PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
(3)          ;*****
(3)          ;*****
(2) 005660 000004          TST3:  SCOPE
(1) 005662 012767 000001 173314    MOV      @1,@TIMES   ;;DO 1 ITERATION
5668          UN:
(1) 005670 012737 000003 004174    MOV      @TTNO,@@TSTNM ;THIS SAVES TEST NUMBER
(1)
5669 005676 013701 001650          MOV      @@RHAS,R1   ;R1 HAS ADDRESS OF RHAS
5670 005702 012711 177777          MOV      @-1,@R1    ;THIS CLEARS RHAS (SURPRISED!)
5671 005706 011137 001126          MOV      @R1,@@BDDAT ;TEST DATA
5672 005712 105737 001126          TSTB   @@BDDAT
5673
```

```
(1) 005716 001405          BEQ     TST4      ;BRANCH IF GOOD
(1)
5674 005720 005037 001124    CLR     000GDDAT   ;GOOD DATA
5675 005724 010137 024234    MOV     R1,00REGADR ;FAILING REG, RHAS
5676 005730 104001          ERROR    1         ;RHAS DOES NOT CLEAR
5677                                     ;WITH ONES
5678
5679                                     ;*****
(3)                                     ;*TEST 4          TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
(3)                                     ;*****
(2) 005732 000004          TST4:   SCOPE
(1) 005734 012767 000001 173242  MOV     01,0TIMES   ;;DO 1 ITERATION
5680 005742 000005          RESET   ;START WITH AN INIT
5681 005744 004737 036424    JSR     PC,000TKINT ;INITILIZE TK
5682 005750 032737 020000 177570  BIT     0SW13,00SWR ;INHIBIT ERROR TYPEOUT?
5683 005756 001030          BNE     48         ;BRANCH IF YES
5684 005760 104400 005766    TYPE    ,,+4       ;;TYPE ASCIZ STRING
(1) 005764 000425          BR      640        ;;GET OVER THE ASCIZ
(1)                                     ;;,ASCIZ      <15><12>/LOOKING AT RHAS = RP04 DRIVES PRESENT /
(1) 006040          640:
5685 006040 013701 001650          48:   MOV     00RHAS,R1   ;R1 HAS ADDR. OF RHAS
5686 006044 013702 001630          MOV     00RHCS2,R2 ;R2 HAS ADDR. OF RHCS2
5687 006050 005012          CLR     0R2        ;CLEAR RHCS2
5688 006052 012700 000010          MOV     00,,R0     ;COUNT
5689 006056 013704 001634          MOV     00RHER1,R4 ;R4 HAS ADDR. OF RHER1
5690 006062 012714 177777          18:   MOV     0-1,0R4    ;MOVE ERRORS INTO RHER1
5691 006066 005212          INC     0R2        ;INCREMENT UNIT NO.
5692 006070 005300          DEC     R0         ;COUNT
5693 006072 001373          BNE     18         ;BRANCH IF 0 NOT DONE
5694 006074 111137 002006          MOVDB  0R1,00TOTALAT ;SAVE TOTAL ATTENTION
5695                                     ;USED IN DRIVE CLEAR TEST
5696 006100 105037 002007          CLRB   00TOTALAT+1 ;CLEAR UPPER BYTE
5697 006104 105711          TSTB   0R1        ;TEST FOR ANY DRIVES PRESENT
5698 006106 001402          BEQ     28         ;IF SOME NOT THERE BRANCH
5699 006110 000167 000436          JMP     XE2        ;NONE THERE
5700 006114 032737 020000 177570  28:   BIT     0SW13,00SWR ;INHIBIT ERROR TYPE OUT?
5701 006122 001402          BEQ     38         ;BRANCH IF NO
5702 006124 000167 000706          JMP     TST5      ;OUT
5703 006130          38:
(1) 006130 104400 006136          TYPE    ,,+4       ;;TYPE ASCIZ STRING
(1) 006134 000412          BR      650        ;;GET OVER THE ASCIZ
(1)                                     ;;,ASCIZ      <15><12>/NO DRIVES=RHAS=0/
(1) 006162          650:
5704 006162 104400 006170          TYPE    ,,+4       ;;TYPE ASCIZ STRING
(1) 006166 000436          BR      660        ;;GET OVER THE ASCIZ
(1)                                     ;;,ASCIZ      <15><12>/WRITING ONES INTO ERROR REGISTER 01 FOR ALL UNIT NUMBERS/
(1) 006264          660:
5705 006264 104400 006272          TYPE    ,,+4       ;;TYPE ASCIZ STRING
(1) 006270 000441          BR      670        ;;GET OVER THE ASCIZ
(1)                                     ;;,ASCIZ      <15><12>/DOES NOT SET ANY BIT IN THE ATTENTION REGISTER 80 ABORT P
(1) 006374          670:
5706 006374 104400 006402          TYPE    ,,+4       ;;TYPE ASCIZ STRING
(1) 006400 000440          BR      680        ;;GET OVER THE ASCIZ
(1)                                     ;;,ASCIZ      <15><12>/TO LOOP ON THIS TEST WITHOUT PRINTOUT SET SWITCHS 13 0 A
```


(1)	006502			6001			
5707	006502	000137	023246		JMP	008EOP	;GO OUT
5708	006506	104400	006514		TYPE	,,+4	;;TYPE ASCIZ STRING
(1)	006512	000410			BR	698	;;GET OVER THE ASCIZ
(1)					;;,ASCIZ	<15><12>/TEST DRIVE 0/	
(1)	006534			6901			
5709	006534	005037	001742		CLR	00UNITS	
5710	006540	012767	000001	173216	MOV	01,NOUNIT	;NO, UNITS PRESENT=1
5711	006546	005037	001762		CLR	00UNIT	
5712	006552			XE21			
5713	006552	012700	000010	201	MOV	00,,R0	;COUNTER
5714	006556	012703	001742		MOV	0UNITS,R3	;POINTER
5715	006562	012723	177777	301	MOV	0-1,(R3)+	;PRESET BLOCK TO ALL ONES
5716	006566	005300			DEC	R0	;COUNT
5717	006570	001374			BNE	30	;BRANCH IF 0 NOT DONE
5718	006572	012703	001742		MOV	0UNITS,R3	;POINTER
5719	006576	005005			CLR	R5	
5720	006600	005037	001764		CLR	00NOUNIT	;NO, OF UNITS PRESENT
5721	006604	012700	000010		MOV	00,,R0	;COUNTER
5722	006610	011137	001170		MOV	0R1,008TMP0	;TEMPORARY STORAGE
5723	006614	006037	001170	401	ROR	008TMP0	;SET CARRY IF ONE IN 0 BIT
5724							
5725	006620	103065			BCC	50	
5726	006622	010577	173002		MOV	R5,0RHCS2	;INSERT UNIT NUMBER
5727	006626	022777	024020	173022	CMP	024020,0RHDT	;IS THIS A DUAL PORT RP04
5728	006634	001450			BEQ	60	;BRANCH IF YES
5729	006636	022777	020020	173012	CMP	020020,0RHDT	;IS THIS A SINGLE PORT RP04
5730	006644	001444			BEQ	60	;BRANCH IF YES
5731	006646	104400	006654		TYPE	,,+4	;;TYPE ASCIZ STRING
(1)	006652	000410			BR	640	;;GET OVER THE ASCIZ
(1)					;;,ASCIZ	<15><12>/UNIT NUMBER /	
(1)	006674			6401			
5732	006674	010546			MOV	R5,-(SP)	;GET READY TO TYPE UNIT NUMBER
5733	006676	104410			TYPDS		
5734	006700	104400	006706		TYPE	,,+4	;;TYPE ASCIZ STRING
(1)	006704	000405			BR	650	;;GET OVER THE ASCIZ
(1)					;;,ASCIZ	/, RHDT= /	
(1)	006720			6501			
5735	006720	017746	172732		MOV	0RHDT,-(SP)	;GET READY TO TYPE RHDT
5736	006724	104402			TYPOC		
5737	006726	104400	006734		TYPE	,,+4	;;TYPE ASCIZ STRING
(1)	006732	000410			BR	660	;;GET OVER THE ASCIZ
(1)					;;,ASCIZ	/ ---NOT AN RP04/	
(1)	006754			6601			
5738	006754	000407			BR	50	;NO RP04 FOUND SO BRANCH
5739	006756	010523		601	MOV	R5,(R3)+	
5740	006760	104400	001215		TYPE	,0CRLF	
5741	006764	010546			MOV	R5,-(SP)	
5742	006766	104410			TYPDS		;TYPE DRIVE NO.
5743	006770	005237	001764		INC	00NOUNIT	
5744	006774	005205		501	INC	R5	
5745	006776	005300			DEC	R0	
5746	007000	001305			BNE	40	
5747	007002	013737	001742	001762	MOV	00UNITS,00UNIT	

```
5748 007010 013737 001764 001766      MOV      @@NUNIT,@@NUNIT      ;SAVE NO. OF UNITS
5749 007016 005337 001766              DEC      @@NUNIT              ;IF NUNIT = 0 THEN ONLY ONE UNIT
5751 007022 005737 001770              TST      @@SELECT             ;IF NUNIT MORE THAN 0 THEN MORE THAN ONE UNIT
5751 007026 001403                      BEQ      TST5 ;BRANCH IF STARTING FROM 200
5753 007030 013737 001772 001762      MOV      @@UNITSL,@@UNIT      ;SET UNIT NUMBER
5761 (3) ;*****
(4) ;TEST 5      TEST SERIAL NUMBER AND DRIVE TYPEI
(4) ;*          READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER
(4) ;*          TYPE IT OUT AND PROCEED
(4) ;*          TO LOOP HERE SET SWITCH 0 AND THIS TEST NO AND RESTART
(3) ;*****
(2) 007036 000004      TST5:  SCOPE
(1) 007040 012767 000001 172136      MOV      @1,@TIMES           ;;DO 1 ITERATION
(1) 007046 012767 007250 172032      MOV      @16,@LPADR          ;;SET SCOPE LOOP ADDRESS
5763 007054 004737 024470      JSR      PC,@@CLDISK         ;FILL UNIT NO.
5763 007060 005037 002004      CLR      @@ATTENT            ;CLEAR
5764 007064 013700 001762      MOV      @@UNIT,@R0          ;R0 CONTAINS UNIT NO
5764 007070 116037 004200 002004      MOV@B   ATABLE(R0),@@ATTENT  ;SET APPROPRIATE ATTENTION BIT
5764 007076 104400 007104      TYPE    ,,+4                 ;;TYPE ASCIZ STRING
(1) 007102 000415      BR      640                   ;;GET OVER THE ASCIZ
(1) ;ASCIZ      <15><12>/TESTING DRIVE NUMBER /
(1) 007136 640:
5767 007136 013746 001762      MOV      @@UNIT,-(SP)        ;UNIT NO. TO STACK
5768 007142 104402              TYPOC                          ;TYPE DRIVE NO.
5769 007144 104400 001215      TYPE    ,@CRLF
5770 007150 104400 007156      TYPE    ,,+4                 ;;TYPE ASCIZ STRING
(1) 007154 000410      BR      650                   ;;GET OVER THE ASCIZ
(1) ;ASCIZ      <15><12>/SERIAL NO. = /
(1) 007176 650:
5771 007176 017746 172456      MOV      @RHSH,-(SP)         ;;SAVE @RHSH FOR TYPEOUT
(1) 007202 104402              TYPOC                          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
5772 007204 104400 001215      TYPE    ,@CRLF
5773 007210 104400 007216      TYPE    ,,+4                 ;;TYPE ASCIZ STRING
(1) 007214 000410      BR      660                   ;;GET OVER THE ASCIZ
(1) ;ASCIZ      <15><12>/DRIVE TYPE = /
(1) 007236 660:
5774 007236 017746 172414      MOV      @RHDT,-(SP)         ;;SAVE @RHDT FOR TYPEOUT
(1) 007242 104402              TYPOC                          ;GO TYPE--OCTAL ASCII(ALL DIGITS)
5775 007244 104400 001215      TYPE    ,@CRLF
5776 007250 005777 172404      TST     @RHSH                ;READ SERIAL NO. AND DRIVE TYPE
5777 007254 005777 172376      TST     @RHDT                ;THESE TWO ARE TO HELP SCOPE LOOPS
5778 007260 017737 172374 002000      MOV     @RHSH,@@SAVSH        ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
5779 007266 017737 172364 001776      MOV     @RHDT,@@SAVDT        ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
5781
5782
5790 (3) ;*****
(4) ;TEST 6      CHECK MOL TO BE LOW
```

```
(4) ;* MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
(4) ;* IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
(4) ;* HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE
(3) ;*****
(2) 007274 000004 TST6: SCOPE
5791
(1) 007276 012737 000006 004174 MOV OTTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
5792 007304 004737 024470 JSR PC,00CLDISK ;GIVE INITILIZE
5793 007310 032713 010000 BIT 0MOL,0R3 ;CHECK MOL IN RHDS1
5794
(1) 007314 001550 BEQ TST7 ;BRANCH IF MOL LOW
(1)
5795 007316 104400 007324 TYPE ,,+4 ;;TYPE ASCIZ STRING
(1) 007322 000421 BR 648 ;;GET OVER THE ASCIZ
(1) ;;.ASCIZ <15><12>/DRIVE IS ON LINE = MOL IS HIGH/
648:
5796 007366 104400 007374 TYPE ,,+4 ;;TYPE ASCIZ STRING
(1) 007372 000424 BR 658 ;;GET OVER THE ASCIZ
(1) ;;.ASCIZ <15><12>/HIT STOP ON DRIVE TO GET IT OFF LINE/
658:
5797 007444 104400 007452 TYPE ,,+4 ;;TYPE ASCIZ STRING
(1) 007450 000430 BR 668 ;;GET OVER THE ASCIZ
(1) ;;.ASCIZ <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS LOW/
668:
5798 007532 032713 010000 181 BIT 0MOL,0R3 ;CHECK MOL IN RHDS1
5799 007536 001375 BNE 18 ;BRANCH IF MOL IS HIGH
5800 007540 104400 007546 TYPE ,,+4 ;;TYPE ASCIZ STRING
(1) 007544 000434 BR 678 ;;GET OVER THE ASCIZ
(1) ;;.ASCIZ <15><12>/GOOD = MOL IS NOW LOW , PROGRAM WILL NOW BE EXECUTED/
678:
5801
5802
5803
5812
5813 ;*****
(3) ;*TEST 7 PACK ACKNOWLEDGE COMMAND TEST
(4)
(4) ;* THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
(4) ;* THEN ALL REGISTERS WILL BE CHECKED
(4) ;* RH CLEAR WILL BE GIVEN
(4) ;* THEN ALL REGISTERS WILL BE CHECKED
(3) ;*****
(2) 007636 000004 TST7: SCOPE
5814 007640 012706 001000 MOV 0STACK,SP ;RESET STACK
5815
(1) 007644 012737 000007 004174 MOV OTTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
5816
5817 007652 004737 024470 JSR PC,00CLDISK ;INIT AND SET UP GENERAL REG.
5818 ;AND UNIT NUMBER
5819 007656 012777 000001 171766 MOV 0DMD,0RHMR ;SET DIAGNOSTIC MODE
```

```

5020
5021 007664 013777 002056 171740      MOV      00PKACK,0RHCS1 ;LOAD PACK ACKNOWLEDGE COMMAND INTO RHCS1
5022
5023                                     ;SAVE REGISTERS FOR COMPARISON AFTER GO
5024 007672 004037 025146      JSR      R0,00SAVER      ;SAVE
5025 007676 001624                                     RHWC      ;FROM
5026 007700 003130                                     REINTO    ;TO
5027 007702 000023      19,          ;NUMBER OF REGISTERS SAVED
5028
5029                                     ;GIVE GO TO PACK ACKNOWLEDGE COMMAND
5030 007704 052777 000001 171720      BIS      0GO,0RHCS1      ;GO TO PACK ACKNOWLEDGE COMMAND
5031
5032                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
5033 007712 052737 000100 003160      BIS      0VV,00REINTO+30 ;SAVED RHDS1
5034
5035                                     ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
5036                                     ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
5037                                     ;BE DONE
5038 007720 004037 025146      JSR      R0,00SAVER      ;SAVE
5039 007724 001624                                     RHWC      ;FROM
5040 007726 002064      WRFROM
5041 007730 000023      19,          ;NUMBER OF REGISTERS SAVED
5042
5043                                     ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5044                                     ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5045                                     ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5046 007732 113737 003155 002111      MOVB    00REINTO+25,00WRFROM+25;SAVE UPPER RHAS
5047
5048
5049                                     ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
5050                                     ;WITH AFTER GO
5051 007740 004037 025342      JSR      R0,00COMPAR     ;COMPARE
5052 007744 003130      REINTO    ;GOOD BUFFER
5053 007746 002064      WRFROM    ;TEST BUFFER
5054 007750 000023      19,      ;NUMBER
5055 007752 007760      18,      ;RETURN FOR ERROR
5056 007754 007760      18,      ;SAME
5057 007756 010000      28,      ;RETURN FOR GOOD COMPARISON
5058 007760 013705 031136      18:      MOV      00ERWORD,R5      ;GETTING READY TO INDEX
5059 007764 060505      ADD      R5,R5          ;DOUBLE ERROR WORD
5060 007766 016537 001622 024234      MOV      RHWC-2(R5),00REGADR ;FAILING REGISTER ADDRESS
5061
5062 007774 104001      ERROR    1              ;IMPROPER REGISTER CHANGE
5063                                     ;AFTER PACK ACKNOWLEDGE COMMAND
5064                                     ;WITH GO IS GIVEN
5065 007776 000207      RTS      PC              ;RETURN TO COMPARISION
5066
5067 010000      20:
5068
5069                                     ;*****
5070 (4)                                     ;*TEST 10 MAKE CURRENT CYLINDER = 0
5071 (4)                                     ;*****
5072 (3) 010000 000004      TST10: SCOPE
5073 (1) 010002 012706 001000      MOV      0STACK,SP      ;RESET STACK

```

```
(1) 010006 004737 024470 JSR PC,00CLDISK ;INIT DRIVE
(1) 010012 012777 000001 171632 MOV 0DMD,0RHMR ;SET DIAGNOSTIC MODE
(1) 010020 004037 027004 JSR R0,00MAKECYL ;SUBROUTINE TO GIVE A SEEK
(1) ;COMMAND FOLOWED BY A INIT
(1) ;THIS SHUOLD CHANGE RHCC
(1) 010024 000000 0 ;CHANGE RHCC TO 0
(1)
5878 ;*****
(3) ;*TEST 11 CONTROL AND STATUS REGISTER 1 BITS 8 AND 9
(4)
(4) ;* WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
(4) ;* TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256
(4) ;* DATA IS THE CONTENTS OF THE TTY READER STATUS REGISTER
(4) ;* THIS WILL USE BITS A16 AND A17 WHEN THERE IS MORE THAN 20K OF MEMORY
(4)
(3) ;*****
(2) 010026 000004 TST111 SCOPE
5879 010030 000167 000310 JMP TST12 ;ON RH70 JUMP TO NEXT TEST
5880 010034 012706 001000 MOV 0STACK,0P ;RESET STACK
5881
5882
(1) 010040 012737 000011 004174 MOV 0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
5883 010046 004037 024406 JSR R0,00CLAREA ;CLEAR SIMULATED DISK
5884 010052 032734 ;WORD DISK ;FROM
5885 010054 033760 ;WORD TOLGAP+16 ;TO
5886 010056 000000 ;WORD 0 ;DATA
5887 ;THESE ARE SETUP FOR DISKLESS USE ONLY
5888 010060 012737 010000 031016 MOV 0FMT22,00CYL;CYLINDER 0
5889
5890 010066 005037 031020 CLR 00SECOTR ;16 BITS PER WORD
5891 010072 005037 031022 CLR 00KEY1 ;SECTOR 0 TRACK 0
5892 010076 005037 031024 CLR 00KEY2 ;KEY1 0
5893 010102 012737 000400 031064 MOV 0256,,00NOWORD ;KEY2 0
5894 010110 012737 000001 031026 MOV 01,00X ;NO OF DATA WORDS
5895 010116 004537 025646 JSR R5,00CRC ;WRITE DATA
5896 010122 031016 CYL ;GO TO CALCULATE CRC
5897 010124 032716 WCRC
5898
5899
5900 ;THESE ARE REGULAR SETUPS
5901
5902
5903 010126 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
5904 010132 012777 177400 171464 MOV 0-256,,0RHWC ;256 DATA WORDS
5905 010140 013777 001136 171460 MOV 000TKS,0RHBA ;STARTING ADDRESS OF WRITE BUFFER
5906 010146 017737 170764 002010 MOV 00TKS,00TMPILL ;TEMPORARY STORAGE OF DATA
5907 010154 005077 171456 CLR 0RHDBT ;SECTOR 0 TRACK 0
5908 010160 012777 010000 171454 MOV 0FMT22,0RHOF ;16 BITS PER WORD FORMAT
5909 010166 005077 171452 CLR 0RHCA ;CYLINDER 0
5910 010172 004737 024524 JSR PC,00CHECKT ;CHECK FOR DVA,RDY,DPR,DRY
5911 010176 013746 002040 MOV 00WRIDAT,-(0P) ;WRITE DATA=60
5912 010202 052716 001400 BIS 0A16|A17,(0P) ;SET HIGH ORDER UNIBUS BITS
5913 010206 012611 MOV (0P)+,0R1 ;FILL RHC81
```

```

5914 010210 052777 000010 171412      DIS      0BAI,0RHCS2      ;SET BUS ADDRESS INHIBIT
5915 010216 005037 001774          CLR      00ERFLG8      ;CLEAR ERROR FLAG
5916 010222 004737 030706          JSR      PC,00COMHD    ;WRITE DATA
5917                                     ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5918                                     ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
5919                                     ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
5920                                     ;AND SYNCB WERE CORRECTLY DETECTED
5921                                     ;DATA IS TO BE CHECKED
5922 010226 004737 024170          JSR      PC,00PUTREG    ;SAVE REGISTERS
5923 010232 005737 001774          TST      00ERFLG8      ;HAS ANY ERRORS OCCURED?
5924 (1) 010236 001042          BNE      TST12      ;;BRANCH IF YES
(1)
5925 010240 013700 002010          NOV      00TMPILL,R0      ;GOOD DATA
5926 010244 012701 032734          NOV      0DISK,R1        ;DATA WRITTEN INTO "DISK"
5927 010250 012702 000400          NOV      0256,,R2        ;COUNTER
5928 010254 012737 000401 031136 101  NOV      0257,,00ERWORD ;FOR ERROR WORD
5929 010262 020021          CMP      R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
5930 010264 001425          BEQ      30              ;BRANCH IF GOOD
5931 010266 013737 002010 001124          NOV      00TMPILL,000GDDAT ;GOOD DATA
5932 010274 014137 001126          NOV      -(R1),000BDDAT ;BAD DATA
5933 010300 160237 031136          SUB      R2,00ERWORD    ;ERROR WORD NO
5934 010304 005737 001774          TST      00ERFLG8      ;ANY ERRORS ALREADY THERE?
5935 010310 001002          BNE      20              ;BRANCH IF YES
5936 010312 104037          ERROR   37              ;ERROR ON WRITE DATA COMMAND
5937                                     ;SEE NEXT ERROR COMMENTS
5938 010314 000401          BR       648             ;BRANCH TO AVOID PRINTING NEXT ERROR
5939 010316 104040          ERROR   40              ;WORD NO GIVES WORD IN ERROR
5940                                     ;ERROR OCCURED WHILE WRITING
5941                                     ;WITH A16 A17 OF RHCS1 SET
5942 010320 005721          648:   TST      (R1)+        ;UNDO -(R1) FOR BAD DATA
5943 010322 013746 177570          NOV      00SWR,-(SP)     ;GET SWITCH SETTING
5944 010326 042716 177177          BIC      0177177,(SP)   ;KEEP ONLY SWITCH 7 AND 8
5945 010332 022726 000200          CMP      0SW07,(SP)+    ;IS 7 SET AND 8 RESET
5946 (1) 010336 001402          BEQ      TST12      ;BRANCH OUT IF YES
(1)
5947 010340 005302          30:   DEC      R2          ;IF NOT COUNT 256 WORDS
5948 010342 001344          BNE      10              ;BRANCH IF 256 NOT DONE
5949
5950
5951
5952
5953
5954
5955
5956
5958
5969
(3)
(4)
(4)
(4)
(4)
;*****
;*TEST 12      DRIVE TIMING ERROR
;*      A READ HEADER AND DATA IS STARTED ON CYLINDER 0, SECTOR
;*      0, TRACK 0, 260 WORDS.  AFTER THE HEADER IS READ IN CORRECTLY
;*      THEN NO SYNC BYTE (DATA SYNC) IS GIVEN,

```

```

(4) ;* THEN NORMAL DIAGNOSTIC CLOCKS AND DIAGNOSTIC
(4) ;* SECTOR CLOCKS ARE GIVEN FOR 24 BYTES.
(4) ;* THEN 536 BYTES OF SECTOR CLOCKS ONLY ARE GIVEN.
(4) ;* THIS IS TO TO BRING SECTOR PULSE UP
(4) ;* THIS SHOULD SET DRIVE TIMING ERROR
(3) ;*****
(2) 010344 000004 TST12: SCOPE
5970 010346 012706 001000 MOV #STACK,SP ;RESET STACK
5971 (1) 010352 012737 000012 004174 MOV #TTNO,#TSTNM ;THIS SAVES TEST NUMBER
(1)
5972
5973
5974 ;THESE ARE TO SETUP FOR DISKLESS USE
5975 010360 012737 010000 031016 MOV #FMT22,#CYL ;16 BITS PER WORD
5976 ;CYLINDER #
5977 010366 005037 031020 CLR #SECOTR ;SECTOR #
5978 ;TRACK #
5979 010372 005037 031022 CLR #KEY1 ;KEY1 = #
5980 010376 005037 031024 CLR #KEY2 ;KEY2 = #
5981 010402 005037 031026 CLR #X ;THIS IS A READ COMMAND
5982 010406 004537 025646 JSR R5,#CRC ;GO TO CALCULATE CRC
5983 010412 031016 CYL
5984 010414 032716 WCRC
5985
5986
5987 ; THESE ARE REGULAR SETUPS
5988 010416 004737 024470 JSR PC,#CLDISK ;SETUP GENERAL REGISTERS
5989 010422 012777 177374 171174 MOV #260,#RHNC ;256 DATA WORDS 4HEADER
5990 010430 012777 003130 171170 MOV #REINTO,#RHBA ;STARTING ADDRESS OF BUFFER
5991 010436 005077 171174 CLR #RHDBT ;TRACK = #
5992 ;SECTOR = #
5993 010442 012777 014000 171172 MOV #FMT22!ECI,#RHOF ;16 BITS PER WORD
5994 ;ECC CORRECTION INHIBITED
5995 010450 005077 171170 CLR #RHCA ;CYLINDER = #
5996 010454 013711 002046 MOV #REFOR,#R1 ;READ HEADER AND DATA = 72
5997
5998 ;SAVE REGISTERS FOR COMPARISON AFTER SIMULATED DRIVE TIMING ERROR
5999 010460 004037 025146 JSR R0,#SAVER ;SAVE
6000 010464 001624 RHNC ;FROM
6001 010466 001674 WC ;TO
6002 010470 000023 19, ;NUMBER
6003
6004 ;NOW GO WILL BE GIVEN. EVERYTHING WILL BE TREATED
6005 ;NORMALLY FOR THE HEADER, BUT WHEN IT IS TIME FOR
6006 ;DATA ONLY, SECTOR CLOCKS WILL BE GIVEN, NO DIAGNOSTIC
6007 ;CLOCKS WILL BE GIVEN, THIS SHOULD BRING SECTOR PULSE HIGH
6008 ;WITHOUT PUTTING READ DOWN HENCE DTE WILL COME UP.
6009 010472 012737 177777 002014 MOV #-1,#TESDTE ;SET DTE TEST
6010 010500 012737 177777 001774 MOV #-1,#ERFLGS ;THIS WILL BRING THE READ HEADER
6011 ;AND DATA PROCESS OUT AFTER THE
6012 ;HEADER HAS BEEN CORRECTLY READ
6013
6014 010506 004737 030706 JSR PC,#COMHD ;READ HEADER AND DATA

```

```

6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029 010512 012701 000030
6030 010516 013700 001652
6031 010522 012710 000001
6032 010526 052710 000012
6033 010532 042710 000012
6034 010536 012702 000007
6035 010542 052710 000002
6036 010546 042710 000002
6037 010552 005302
6038 010554 001372
6039 010556 005301
6040 010560 001362
6041
6042
6043 010562 012701 001030
6044 010566 052710 000010
6045 010572 042710 000010
6046 010576 005301
6047 010600 001372
6048
6049
6050
6051
6052 010602 012737 177400 001674
6053 010610 012737 003140 001676
6054 010616 052737 140000 001702
6055 010624 052737 010000 001704
6056 010632 012737 000401 001722
6057 010640 052737 140000 001724
6058 010646 012737 000100 001736
6059 010654 012737 000001 001706
6060 010662 013737 002004 001720
6061
6062
6063
6064 010670 004037 025146
6065 010674 001624
6066 010676 002064
6067 010700 000023
6068

```

```

;NOW THE HEADER HAS BEEN READ
;NOW 560 SECTOR CLOCKS WILL BE GIVEN
;GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
;GAP 2 BYTES, TOLERANCE 20 BYTES, EXTRA 2

```

```

;THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
;24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC CLOCKS
;AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC CLOCKS

```

```

;THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC CLOCKS
MOV 024,,R1 ;COUNTER,
MOV 00RHMR,R0 ;GER RHMR ADDRESS
MOV 00DMD,0R0 ;SET DIAGNOSTIC MODE
10: BIS 00NSTCK|NCLK,0R0 ;SET SECTOR CLOCK AND CLOCK
BIC 00NSTCK|NCLK,0R0 ;CLEAR SECTOR CLOCK AND CLOCK
MOV 07,R2 ;COUNTER FOR DIAGNOSTIC CLOCKS
40: BIS 0NCLK,0R0 ;SET CLOCK
BIC 0NCLK,0R0 ;CLEAR CLOCK
DEC R2 ;COUNT TO 7
BNE 40 ;BRANCH IF 7 NOT DONE
DEC R1 ;COUNT
BNE 10 ;BRANCH IF 24 NOT DONE

```

```

;THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC CLOCKS
MOV 0536,,R1 ;COUNTER,
50: BIS 0NSTCK,0R0 ;SET SECTOR CLOCK
BIC 0NSTCK,0R0 ;CLEAR SECTOR CLOCK
DEC R1 ;COUNT
BNE 50 ;BRANCH IF 536 NOT DONE

```

```

;NOW DTE SHOULD BE SET CHANGE SAVED REGISTERS TO EXPECTED VALUE
MOV 0-256,,00WC ;SAVED RHWC
MOV 00REINT0+<4,02>,00BA ;SAVED RHBA
BIS 00SCITRE,00CS1 ;SAVED RHCS1
BIS 0DTE,00ER1 ;SAVED RHER1
MOV 0401,00NR ;SAVED RHNR
BIS 0ATA|ERR,00DS1 ;SAVED RHDS1
MOV 0100,00LA ;SAVED RHLA
MOV 01,00DST ;SAVED RHDST
MOV 00ATTENT,00AS ;SAVED RHAS

```

```

;NOW SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN BE DONE
JSR R0, 00SAVER ;SAVE
RHWC ;FROM
NRFROM ;TO
19. ;NUMBER

```



```
6069 ;FOR RHAS UPPER BYTE
6070 010702 113737 001721 002111 MOVB 00AS+1,00WRFROM+25 ;UPPER RHAS
6071
6072 ;COMPARE HEADER READ
6073 010710 004037 025342 JSR R0,00COMPAR ;COMPARE
6074 010714 031016 CYL ;GOOD BUFFER
6075 010716 003130 REINTO ;TEST BUFFER
6076 010720 000004 4, ;NUMBER
6077 010722 010730 68 ;RETURN FOR ERROR
6078 010724 010730 68 ;SAME
6079 010726 010734 78 ;RETURN FOR GOOD COMPARISON
6080 010730 104010 681 ERROR 10 ;HEADER READ IN DURING THIS TEST IS
6081 ;IN ERROR
6082
6083 010732 000207 RTS PC ;RETURN
6084
6085 010734 781 ;GOOD
6086
6087
6088 ;COMPARE REGISTERS BEFORE COMMAND WITH AFTER COMMAND
6089 010734 004037 025342 JSR R0,00COMPAR ;COMPARE
6090 010740 001674 WC ;GOOD BUFFER
6091 010742 002064 WRFROM ;TEST BUFFER
6092 010744 000022 10, ;NUMBER
6093 010746 010754 28 ;RETURN FOR ERROR
6094 010750 010754 28 ;SAME
6095 010752 010774 38 ;RETURN FOR GOOD COMPARISON
6096
6097 010754 013705 031136 281 MOV 00ERWORD,R5 ;GETTING READY TO INDEX
6098 010760 060505 ADD R5,R5 ;DOUBLE ERROR WORD
6099 010762 016537 001622 024234 MOV RHWC-2(R5),00REGADR ;FAILING REGISTER
6100 010770 104001 ERROR 1 ;IMPROPER REGISTER
6101 ;CHANGE WHILE EXPECTING
6102 ;DTE ERROR
6103 010772 000207 RTS PC ;RETURN
6104
6105 010774 381 ;GOOD
6106
6107
6119
6120 ;*****
(3) ;TEST 13 DRIVE TIMING ERROR
(4)
(4) ;* A WRITE DATA COMMAND IS STARTED ON CYLINDER 0, SECTOR
(4) ;* 0, TRACK 0, 256 WORDS, AFTER THE HEADER IS READ IN CORRECTLY
(4) ;* THEN NO SYNC BYTE (DATA SYNC) IS GIVEN,
(4) ;* THEN NORMAL DIAGNOSTIC CLOCKS AND DIAGNOSTIC
(4) ;* SECTOR CLOCKS ARE GIVEN FOR 24 BYTES,
(4) ;* THEN 536 BYTES OF SECTOR CLOCKS ONLY ARE GIVEN,
(4) ;* THIS IS TO TO BRING SECTOR PULSE UP
(4) ;* THIS SHOULD SET DRIVE TIMING ERROR
(3) ;*****
(2) 010774 000004 TST13: SCOPE
6121 010776 012706 001000 MOV 00STACK,SP ;RESET STACK
```

```

6122 (1) 011002 012737 000013 004174      MOV      @TTNO,@TSTNM      ;THIS GAVES TEST NUMBER
6123 (1)
6124
6125 ;THESE ARE TO SETUP FOR DISKLESS USE
6126 011010 012737 010000 031016      MOV      @FMT22,@CYL      ;16 BITS PER WORD
6127 ;CYLINDER 0
6128 011016 005037 031020      CLR      @SECOTR          ;SECTOR 0
6129 ;TRACK 0
6130 011022 005037 031022      CLR      @KEY1           ;KEY1 = 0
6131 011026 005037 031024      CLR      @KEY2           ;KEY2 = 0
6132 011032 012737 177777 031026      MOV      @-1,@X          ;THIS IS A WRITE DATA COMMAND
6133 011040 004537 025646      JSR      RS,@CRC         ;GO TO CALCULATE CRC
6134 011044 031016
6135 011046 032716
6136
6137
6138 ; THESE ARE REGULAR SETUPS
6139 011050 004737 024470      JSR      PC,@CLDISK      ;SETUP GENERAL REGISTERS
6140 011054 012777 177400 170542      MOV      @-256,@RHWC     ;256 DATA WORDS
6141 011062 012777 002064 170536      MOV      @WRFRON,@RHBA   ;STARTING ADDRESS OF BUFFER
6142 011070 005077 170542      CLR      @RNDST         ;TRACK = 0
6143 ;SECTOR = 0
6144 011074 012777 010000 170540      MOV      @FMT22,@RHOF    ;16 BITS PER WORD
6145 ;ECC CORRECTION INHIBITED
6146 011102 005077 170536      CLR      @RNCA          ;CYLINDER = 0
6147 011106 013711 002040      MOV      @WRIDAT,@R1     ;WRITE DATA = 60
6148
6149 ;SAVE REGISTERS FOR COMPARISON AFTER SIMULATED DRIVE TIMING ERROR
6150 011112 004037 025146      JSR      RS,@SAVER      ;SAVE
6151 011116 001624      RHWC          ;FROM
6152 011120 001674      WC           ;TO
6153 011122 000023      19.         ;NUMBER
6154
6155 ;NOW GO WILL BE GIVEN, EVERYTHING WILL BE TREATED
6156 ;NORMALLY FOR THE HEADER, BUT WHEN IT IS TIME FOR
6157 ;DATA ONLY, SECTOR CLOCKS WILL BE GIVEN, NO DIAGNOSTIC
6158 ;CLOCKS WILL BE GIVEN, THIS SHOULD BRING SECTOR PULSE HIGH
6159 ;WITHOUT PUTTING READ DOWN HENCE DTE WILL COME UP.
6160 011124 012737 177777 002014      MOV      @-1,@TESDTE     ;SET DTE TEST
6161 011132 012737 177777 001774      MOV      @-1,@ERFLG8     ;THIS WILL BRING THE READ HEADER
6162 ;AND DATA PROCESS OUT AFTER THE
6163 ;HEADER HAS BEEN CORRECTLY READ
6164
6165 011140 004737 030706      JSR      PC,@COMHD       ;READ HEADER AND DATA
6166
6167 ;NOW THE HEADER HAS BEEN READ
6168 ;NOW 560 SECTOR CLOCKS WILL BE GIVEN
6169 ;GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
6170 ;GAP 2 BYTES, TOLERANCE 20 BYTES, EXTRA 2
6171
6172 ;THESE 560 SECTOR CLOCKS ARE DIVIDED INTO TWO GROUPS
6173 ;24 SECTOR CLOCKS WITH NORMAL DIAGNOSTIC CLOCKS

```

```

6174                                     ;AND 536 SECTOR CLOCKS WITHOUT ANY DIAGNOSTIC CLOCKS
6175
6176
6177
6178
6179
6180 011144 012701 000030                ;THIS GIVES 24 SECTOR CLOCKS WITH DIAGNOSTIC CLOCKS
6181 011150 013700 001652                MOV     #24,,R1 ;COUNTER,
6182 011154 012710 000001                MOV     @RHMR,R0 ;GER RHMR ADDRESS
6183 011160 052710 000012                MOV     @DMD,@R0 ;SET DIAGNOSTIC MODE
6184 011164 042710 000012                10:    BIS     @MSTCK|MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
6185 011170 012702 000007                BIC     @MSTCK|MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
6186 011174 052710 000002                MOV     #7,R2 ;COUNTER FOR DIAGNOSTIC CLOCKS
6187 011200 042710 000002                40:    BIS     @MCLK,@R0 ;SET CLOCK
6188 011204 005302                        BIC     @MCLK,@R0 ;CLEAR CLOCK
6189 011206 001372                        DEC     R2 ;CUOUNT TO 7
6190 011210 005301                        BNE     #0 ;BRANCH IF 7 NOT DONE
6191 011212 001362                        DEC     R1 ;COUNT
6192
6193                                     ;THIS GIVES 536 SECTOR CLOCKS WITHOUT DIAGNOSTIC CLOCKS
6194 011214 012701 001030                MOV     #536,,R1 ;COUNTER,
6195 011220 052710 000010                50:    BIS     @MSTCK,@R0 ;SET SECTOR CLOCK
6196 011224 042710 000010                BIC     @MSTCK,@R0 ;CLEAR SECTOR CLOCK
6197 011230 005301                        DEC     R1 ;COUNT
6198 011232 001372                        BNE     #0 ;BRANCH IF 536 NOT DONE
6199
6200
6201
6202                                     ;ECC PATTERN REGISTER IS NOT CHECKED
6203 011234 017737 170424 001734        MOV     @RHEC2,@EC2 ;RHEC2 IS NOT CHECKED
6204
6205
6206                                     ;NOW DTE SHOULD BE SET CHANGE SAVED REGISTERS TO EXPECTED VALUE
6207 011242 012737 177416 001674        MOV     #-242,,@WC ;SAVED RHWC
6208 011250 012737 002120 001676        MOV     @NRFROM+<14,*2>,@BA ;SAVED RHBA
6209 011256 052737 140000 001702        BIS     @SCITRE,@CS1 ;SAVED RHCS1
6210 011264 052737 000300 001700        BIS     @OR|IR,@CS2 ;SAVED RHCS2
6211 011272 052737 010000 001704        BIS     @DTE,@ER1 ;SAVED RHER1
6212 011300 012737 000201 001722        MOV     @DENVL|DMD,@MR ;SAVED RHMR
6213 011306 052737 140000 001724        BIS     @ATA|ERR,@DS1 ;SAVED RHDS1
6214 011314 012737 000100 001736        MOV     #100,@LA ;SAVED RHLA
6215 011322 012737 000001 001706        MOV     #1,@DST ;SAVED RHDST
6216 011330 013737 002004 001720        MOV     @ATTENT,@AS ;SAVED RHAS
6217
6218
6219                                     ;NOW SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN BE DONE
6220 011336 004037 025146                JSR     R0, @SAVER ;SAVE
6221 011342 001624                        RHWC ;FROM
6222 011344 003130                        REINTO ;TO
6223 011346 000023                        19. ;NUMBER
6224
6225                                     ;FOR RHAS UPPER BYTE
6226 011350 113737 001721 003155        MOVB   @AS+1,@REINTO+25 ;UPPER RHAS
6227

```

```
6228
6229
6230 ;COMPARE REGISTERS BEFORE COMMAND WITH AFTER COMMAND
6231 011356 004037 025342 JSR R0,00COMPAR ;COMPARE
6232 011362 001674 WC ;GOOD BUFFER
6233 011364 003130 REINTO ;TEST BUFFER
6234 011366 000022 10, ;NUMBER
6235 011370 011376 20 ;RETURN FOR ERROR
6236 011372 011376 20 ;SAME
6237 011374 011416 30 ;RETURN FOR GOOD COMPARISON
6238
6239 011376 013705 031136 20: MOV 00ERWORD,R5 ;GETTING READY TO INDEX
6240 011402 060505 ADD R5,R5 ;DOUBLE ERROR WORD
6241 011404 016537 001622 024234 NOV RHWC-2(R5),00REGADR ;FAILING REGISTER
6242 011412 104001 ERROR 1 ;IMPROPER REGISTER
6243 ;CHANGE WHILE EXPECTING
6244 ;DTE ERROR
6245 011414 000207 RTS PC ;RETURN
6246
6247 011416 30: ;GOOD
6248
6249
6259
6260 ;*****
(3) ;*TEST 14 DRIVE TIMING ERROR
(4)
(4) ;* A WRITE HEADER AND DATA COMMAND IS GIVEN
(4) ;* TO CYLINDER 0, TRACK 0, 256, WORDS
(4) ;* AFTER SECTOR IS FOUND (THE SECTOR FOUND FLOP IS HIGH)
(4) ;* NO MORE DIAGNOSTIC CLOCKS ARE GIVEN,
(4) ;* ONLY SECTOR CLOCKS ARE GIVEN TILL SECTOR PULSE IS HIGH
(4) ;* THIS SHOULD SET DRIVE TIMING ERROR
(3) ;*****
(2) 011416 000004 TST14: SCOPE
6261 011420 012706 001000 NOV 0STACK,SP ;RESET STACK
6262 (1) 011424 012737 000014 004174 NOV 0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
6263
6264 ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
6265
6266
6267 011432 012737 010000 034122 MOV 0FMT22,00WCYL ;FORMAT 22=16 BITWORDS AND
6268 ;CYLINDER 0
6269 011440 005037 034124 CLR 00WSECTR ;TRACK=0, SECTOR=0
6270 011444 005037 034126 CLR 00WKEY1 ;KEY1=0
6271 011450 005037 034130 CLR 00WKEY2 ;KEY2=0
6272 011454 012737 000400 034162 NOV 0256,,00FNWORD ;256 DATAWORDS
6273 011462 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
6274 011466 034122 WCYL
6275 011470 034132 GCRC
6276
6277
6278 ; THESE ARE REGULAR SETUPS
```

6279	011472	004737	024470		JSR	PC,00CLDISK	;SETUP GENERAL REGISTERS
6280	011476	012777	177374	170120	MOV	0-260,,0RHWC	;256 DATA WORDS 4HEADER
6281	011504	012777	002064	170114	MOV	0WRFRON,0RHBA	;STARTING ADDRESS OF BUFFER
6282	011512	005077	170120		CLR	0RHDS1	;TRACK = 2
6283							;SECTOR = 0
6284	011516	012777	014000	170116	MOV	0FMT22IECI,0RHOF	;16 BITS PER WORD
6285							;ECC CORRECTION INHIBITED
6286	011524	005077	170114		CLR	0RHCA	;CYLINDER = 0
6287	011530	013711	002042		MOV	00WRIFOR,0R1	;WRITE HEADER AND DATA = 62
6288							
6289							;SAVE REGISTERS FOR COMPARISON AFTER SIMULATED DRIVE TIMING ERROR
6290	011534	004037	025146		JSR	R0,00SAVER	;SAVE
6291	011540	001624			RHWC		;FROM
6292	011542	001674			WC		;TO
6293	011544	000023			19,		;NUMBER
6294							
6295							;NOW GO WILL BE GIVEN, EVERYTHING WILL BE TREATED
6296							;NORMALLY TILL HEADER IS TO BE GIVEN, THEN ONLY
6297							;SECTOR CLOCKS WILL BE GIVEN, NO DIAGNOSTIC
6298							;CLOCKS WILL BE GIVEN, THIS SHOULD BRING SECTOR PULSE HIGH
6299							;WITHOUT PUTTING READ DOWN HENCE DTE WILL COME UP.
6300	011546	012737	177777	002014	MOV	0-1,00TESDTE	;SET DTE TEST
6301	011554	012737	177777	001774	MOV	0-1,00ERFLG0	;THIS WILL BRING THE READ HEADER
6302							;AND DATA PROCESS OUT AFTER THE
6303							;HEADER HAS BEEN CORRECTLY READ
6304							
6305	011562	004737	033776		JSR	PC,00CONWHD	;WRITE HEADER AND DATA
6306							
6307							;NOW SECTOR HAS BEEN FOUND
6308							;NOW 609 SECTOR CLOCKS WILL BE GIVEN
6309							;39 BYTES FOR SECTOR GAP
6310							;1 BYTE FOR HEADER SYNC
6311							;0 BYTES FOR HEADER
6312							;GAP 11 BYTES, SYNC 1 BYTE, DATA 512, ECC 4 BYTES
6313							;GAP 2 BYTES, TOLERANCE 20 BYTES, EXTRA 3
6314							
6315							
6316							
6317							;THIS GIVES 609 SECTOR CLOCKS WITHOUT DIAGNOSTIC CLOCKS
6318	011566	012701	001141		MOV	0609,,R1	;COUNTER.
6319	011572	052710	000010	501	BIS	0NSTCK,0R0	;SET SECTOR CLOCK
6320	011576	042710	000010		BIC	0NSTCK,0R0	;CLEAR SECTOR CLOCK
6321	011602	005301			DEC	R1	;COUNT
6322	011604	001372			BNE	50	;BRANCH IF 536 NOT DONE
6323							
6324							
6325							
6326							;NOW DTE SHOULD BE SET CHANGE SAVED REGISTERS TO EXPECTED VALUE
6327	011606	012737	177404	001674	MOV	0-252,,00WC	;SAVED RHWC
6328	011614	012737	002104	001676	MOV	0WRFRON+<0,02>,00BA	;SAVED RHBA
6329	011622	052737	140000	001702	BIS	0SCITRE,00CS1	;SAVED RHCS1
6330	011630	042737	000100	001700	BIC	0IR,00CS2	;SAVED RHCS2
6331	011636	052737	000200	001700	BIS	0OR,00CS2	;SAVED RHCS2
6332	011644	052737	010000	001704	BIS	0DTE,00ER1	;SAVED RHER1

```
6333 011652 012737 000401 001722      MOV      0401,00MR      ;SAVED RHMR
6334 011660 052737 140000 001724      BIS      0ATAIERR,00DS1 ;SAVED RHDS1
6335 011666 012737 000100 001736      MOV      0100,00LA      ;SAVED RHLA
6336 011674 012737 000001 001706      MOV      01,00DST       ;SAVED RHDST
6337 011702 013737 002004 001720      MOV      00ATTENT,00AS  ;SAVED RHAS
6338
6339
6340
6341 011710 004037 025146      ;NOW SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN BE DONE
6342 011714 001624      JSR      R0,      00SAVER ;SAVE
6343 011716 003130      RHWC      ;FROM
6344 011720 000023      REINTO      ;TO
6345
6346
6347 011722 113737 001721 003155      ;FOR RHAS UPPER BYTE
6348      MOVB     00AS+1,00REINTO+25 ;UPPER RHAS
6349
6350
6351
6352 011730 004037 025342      ;COMPARE REGISTERS BEFORE COMMAND WITH AFTER COMMAND
6353 011734 001674      JSR      R0,00COMPAR    ;COMPARE
6354 011736 003130      WC      ;GOOD BUFFER
6355 011740 000022      REINTO      ;TEST BUFFER
6356 011742 011750      10,      ;NUMBER
6357 011744 011750      20      ;RETURN FOR ERROR
6358 011746 011770      20      ;SAME
6359      30      ;RETURN FOR GOOD COMPARISON
6360 011750 013705 031136      201      MOV      00ERWORD,R5      ;GETTING READY TO INDEX
6361 011754 060505      ADD      R5,R5          ;DOUBLE ERROR WORD
6362 011756 016537 001622 024234      MOV      RHWC-2(R5),00REGADR ;FAILING REGISTER
6363 011764 104001      ERROR    1             ;INPROPER REGISTER
6364
6365
6366 011766 000207      RTS      PC             ;CHANGE WHILE EXPECTING
6367
6368 011770      ;DTE ERROR
6369
6370
6371
6382
6383 011770 000004      ;RETURN
6384
6385
;*****
;TEST 15 SECTOR SELECTION
; THE SECTOR SELECTION LOGIC IS CHECKED HERE
; EACH SECTOR ON TRACK ZERO IS WRITTEN INTO
; DATA IS - 19 WORDS OF ZEROS - SYNC WORDS, 4 HEADER WORDS
; 1 CRC WORD, 5 WORDS OF ZEROS, 1 SYNC WORD, 100 ZEROS
; (DATA), 1 SYNC WORD, 70 SECTOR NUMBER TO VARY
;
; THE WRITTEN DATA IS CHECKED IN MEMORY
;*****
TST15: SCOPE
6383 011772 012706 001000      MOV      0STACK,SP      ;RESET STACK
6384
6385
```

```

(1) 011776 012737 000015 004174      MOV      @TTNO,@TSTNM      ;THIS SAVES TEST NUMBER
(1)
6386
6387 012004 012737 000026 002016      MOV      @22,,@TAGDTE      ;22 SECTORS
6388                                     ;THIS TEST REPEATS
6389                                     ;ITSELF 22 TIMES
6390
6391                                     ;THE FOLLOWING INITIALIZES FOR SECTOR 0
6392
6393 012012 005037 012122      CLR      @0553+2          ;HEADER (SECTOR)
6394 012016 012737 000025 012126      MOV      @21,,@0554+2      ;HEADER (KEY1)
6395 012024 012737 000025 012132      MOV      @21,,@0555+2      ;HEADER (KEY2)
6396 012032 005037 012160      CLR      @0557+2          ;DATA (SECTOR)
6397 012036 005037 012242      CLR      @0558+2          ;DATA
6398 012042 005037 012272      CLR      @0559+2          ;SECTOR (SIMULATED DISK)
6399 012046 012737 000025 012300      MOV      @21,,@055A+2      ;KEY1 (SIMULATED DISK)
6400 012054 012737 000025 012306      MOV      @21,,@055B+2      ;KEY2 (SIMULATED DISK)
6401 012062 005037 012346      CLR      @055C+2          ;SECTOR (RHDST)
6402
6403                                     ;CLEAR SIMULATED DISK AREA
6404 012066                                     551:
6405 012066 012700 032636      18:  MOV      @SECGAP,R0      ;POINTER
6406 012072 012701 000460      MOV      @304,,R1          ;COUNTER
6407 012076 005020      28:  CLR      (R0)+            ;CLEAR SIMULATED DISK AREA
6408 012100 005301      DEC      R1                ;COUNT
6409 012102 001375      BNE     28
6410
6411                                     ;SETUP GENERAL REGISTERS
6412 012104 004737 024470      JSR     PC,@CLDISK
6413
6414                                     ;SETUP WRITE FROM BUFFER
6415 012110 012700 002064      MOV     @WRFROM,R0
6416
6417                                     ;HEADER
6418 012114 012720 010000      MOV     @FMT22,(R0)+      ;FORMAT 16 BITS PER WORD
6419                                     ;CYLINDER 0
6420 012120 012720 000000      553:  MOV     @0,(R0)+          ;SECTOR TO VARY
6421 012124 012720 000025      554:  MOV     @21,,(R0)+        ;KEY1 TO VARY
6422 012130 012720 000025      555:  MOV     @21,,(R0)+        ;KEY2 TO VARY
6423
6424                                     ;DATA IN WRITE FROM BUFFER ALTHOUGH THIS IS DATA AND NOT
6425                                     ;HEADER THE SECTOR WITH SYNC BYTES WILL BE GIVEN AS DATA
6426                                     ;DATA IS - 19 WORDS OF ZEROS - SYNC WORDS, 4 HEADER WORDS
6427                                     ;1 CRC WORD, 5 WORDS OF ZEROS, 1 SYNC WORD, 100 ZEROS
6428                                     ;(DATA), 1 SYNC WORD, 70 SECTOR NUMBER TO VARY
6429
6430 012134 012705 000023      MOV     @19,,R5            ;COUNTER
6431 012140 005020      68:  CLR     (R0)+            ;19 ZEROS
6432 012142 005305      DEC     R5                ;COUNT
6433 012144 001375      BNE     68                ;19 DONE?
6434 012146 013720 031120      MOV     @@RSYNC,(R0)+      ;SYNC = 14400
6435 012152 012720 010000      MOV     @FMT22,(R0)+      ;CYLINDER 0
6436 012156 012720 000000      557:  MOV     @0,(R0)+          ;SECTOR TO VARY
6437 012162 005020      CLR     (R0)+

```

6438	012164	005020			CLR	(R0)+	
6439	012166	004537	025646		JSR	R5,00CRC	;CALCULATE CRC FOR ABOVE 4 WORDS
6440	012172	002134			WRFROM+50		;4 WORDS START FROM HERE
6441	012174	002144			WRFROM+60		;PUT CRC HERE
6442							
6443	012176	005720			TST	(R0)+	;INCREMENT R0
6444							
6445	012200	012705	000005		MOV	05,,R5	
6446	012204	005020		001	CLR	(R0)+	;5 WORDS OF ZEROS
6447	012206	005305			DEC	R5	;COUNT
6448	012210	001375			BNE	00	;BRANCH IF 5 NOT DONE
6449							
6450	012212	013720	031120		MOV	00RBYNC,(R0)+	;SYNC = 14400
6451							
6452	012216	012705	000144		MOV	0100,,R5	
6453	012222	005020		901	CLR	(R0)+	;100 WORDS OF ZEROS
6454	012224	005305			DEC	R5	
6455	012226	001375			BNE	90	
6456							
6457	012230	013720	031120		MOV	00RBYNC,(R0)+	;SYNC = 14400
6458	012234	012705	000106		MOV	070,,R5	
6459	012240	012720	000000	00101	MOV	00,(R0)+	;SECTOR TO VARY
6460	012244	005305			DEC	R5	
6461	012246	001374			BNE	0010	
6462							
6463							;CLEAR REST OF 256 WORDS THAT IS 54 WORDS OF ZEROS
6464							
6465	012250	012705	000066		MOV	054,,R5	
6466	012254	005020		1101	CLR	(R0)+	
6467	012256	005305			DEC	R5	
6468	012260	001375			BNE	110	
6469							
6470							;THESE ARE TO BE SET UP FOR DISKLESS USE ONLY
6471	012262	012737	010000	034122	MOV	0FMT22,00WCYL	;FORMAT = 16 BIT WORDS
6472							;CYLINDER = 0
6473	012270	012737	000000	034124	00121	MOV	00,00WSECTR
6474	012276	012737	000025	034126	00131	MOV	021,,00WKEY1
6475	012304	012737	000025	034130	00141	MOV	021,,00WKEY2
6476	012312	012737	000312	034162		MOV	0202,,00FNWORD
6477	012320	004537	025646		JSR	R5,00CRC	;CALCULATE CRC
6478	012324	034122			WCYL		;FIRST WORD
6479	012326	034132			GCRC		;PUT HERE
6480							
6481							;THESE ARE REGULAR SETUPS
6482	012330	012777	177400	167266	MOV	0-256,,0RHWC	;202 DATA, 4 HEADER
6483	012336	012777	002064	167262	MOV	0WRFROM,0RHBA	;FILL BUS ADDRESS
6484	012344	012777	000000	167264	00151	MOV	00,0RHDS1
6485	012352	013777	002042	167252	MOV	00WRIFOR,0RHCS1	;GET READY TO DO
6486							;WRITE HEADER AND DATA
6487							;WITH 62 IN RHCS1
6488	012360	012777	010000	167254	MOV	0FMT22,0RHOF	;16 BITS PER WORD FORMAT
6489	012366	005077	167252		CLR	0RHCA	;CYLINDER = 0
6490							
6491	012372	005037	001774		CLR	00ERFLG0	;CLEAR ERROR FLAG


```

6492
6493
(1)
(1) 012376 004767 012122 JSR PC,CHECKT ;CHECK DVA, RDY, DPR, DRY
(1)
6494
6495 012402 004737 033776 JSR PC,00COMWHD ;WRITE HEADER AND DATA COMMAND
6496 012406 005737 001774 TST 00ERFLG8 ;HAS ANY ERRORS OCCURRED
6497
(1) 012412 001046 BNE TST16 ;BRANCH IF YES
(1)
6498
6499 012414 004737 024706 JSR PC,00CHECKE ;CHECK DVA,RDY,DRY,DPR
6500
6501 ;NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER
6502 012420 004037 025342 JSR R0,00COMPAR ;CHECK
6503 012424 002074 WRFROM+0, ;GOOD BUFFER
6504 012426 032734 DISK ;TEST BUFFER
6505 012430 000400 256. ;NUMBER OF WORDS
6506 012432 012440 160 ;RETURN POINT FOR ERROR HEADER
6507 012434 012444 170 ;RETURN POINT FOR ERROR DATA
6508 012436 012450 180 ;RETURN FOR GOOD COMPARISON
6509 012440 104007 160: ERROR 7
6510 012442 000207 RTS PC
6511 012444 104010 170: ERROR 10
6512 012446 000207 RTS PC
6513
6514 ;THE FOLLOWING INCREMENTS ARE TO CHANGE THE ABOVE SET UP
6515 ;TO WRITE ON THE NEXT SECTOR
6516
6517 012450 005237 012122 100: INC 00883+2 ;HEADER (SECTOR)
6518 012454 005337 012126 DEC 00884+2 ;HEADER (KEY1)
6519 012460 005337 012132 DEC 00885+2 ;HEADER (KEY2)
6520 012464 005237 012160 INC 00887+2 ;DATA (SECTOR)
6521 012470 005237 012242 INC 008810+2 ;DATA
6522 012474 005237 012272 INC 008812+2 ;SECTOR (SIMULATED DISK)
6523 012500 005337 012300 DEC 008813+2 ;KEY1 (SIMULATED DISK)
6524 012504 005337 012306 DEC 008814+2 ;KEY2 (SIMULATED DISK)
6525 012510 005237 012346 INC 008815+2 ;SECTOR (RHDST)
6526
6527 012514 005337 002016 882: DEC 00TAGDTE ;COUNT DOWN FOR 22 SECTORS
6528 012520 001001 BNE 18 ;BRANCH IF 22 SECTORS NOT DONE
6529
(1) 012522 000402 BR TST16 ;DGO OUT
(1)
6530 012524 000137 012066 10: JMP 00881 ;GO BACK
6531
6532
6533
6542
6543
(3) ;*****
(4) ;*TEST 16 WRITE ECC TEST 1
(4) ;* THIS IS A WRITE ECC TEST

```

```

(4) ;* WRITE CYLINDERS, FORMAT 16 BITS PER WORD
(4) ;* TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
(4) ;* OF ALL ZEROS.
(4) ;*****
(3) TST16: SCOPE
(2) 012530 000004
6544 012532 012706 001000 MOV #STACK,SP ;RESET STACK
6545
6546
(1) 012536 012737 000016 004174 MOV #TTNO,#TSTNM ;THIS SAVES TEST NUMBER
(1)
6547 012544 012700 032636 MOV #SECGAP,R0 ;POINTER
6548 012550 012701 000402 MOV #250,,R1 ;COUNTER
6549 012554 012720 177777 18: MOV #-1,(R0)+ ;FILL SIMULATOR DISK WITH ONES
6550 012560 005301 DEC R1
6551 012562 001374 BNE 18
6552 012564 004767 011700 JSR PC,CLDISK ;THIS IS USED TO SET GENERAL REGISTERS
6553
6554 ;THESE ARE FOR ECC TEST ONLY
6555
6556 012570 012737 177777 002012 MOV #-1,#TSECC ;THIS IS AN ECC TEST
6557 012576 005037 027140 CLR #POSITI ;CLEAR ERROR POSITION COUNTER
6558 012602 013737 027134 027136 MOV #NCODE,#NCOUNT ;TEMPORARY N-CODE COUNTER
6559 012610 013737 027142 027150 MOV #HARDER,#HADTMP ;TEMPORARY HARD ERROR COUNTER
6560 012616 005037 027126 CLR #GECC1 ;ECC LOW ORDER TO BE GENERATED
6561 012622 005037 027130 CLR #GECC2 ;ECC HIGH ORDER TO BE GENERATED
6562 012626 005037 027144 CLR #DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
6563 012632 005037 027146 CLR #ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
6564
6565
6566
6567
6568 ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
6569
6570 012636 012737 010000 034122 MOV #FMT22,#WCYL ;FORMAT22=16BIT WORDS AND
6571 ;CYLINDER 0
6572 012644 012737 000001 034124 MOV #1,#NSECTR ;TRACK=0, SECTOR=1
6573 012652 005037 034126 CLR #WKEY1 ;KEY1=0
6574 012656 005037 034130 CLR #WKEY2 ;KEY2=0
6575 012662 012737 000400 034162 MOV #256,,#FNWORD ;256 DATA WORDS
6576 012670 004537 025646 JSR R5,#CRC ;GO TO CALCULATE CRC
6577 012674 034122 WCYL
6578 012676 034132 GCRC
6579
6580 ;THESE ARE REGULAR SETUPS
6581
6582 012700 012777 177374 166716 MOV #-260,,#RHWC ;256 DATA WORDS 4 HEADER WORDS
6583 012706 012700 002064 MOV #WFRON,R0 ;THESE TWO INSTRUCTIONS GETS
6584 012712 010077 166710 MOV R0,#RHBA ;ADDR. OF WFRON INTO R0 AND
6585 ;BUS ADDRESS REGISTER
6586 012716 012720 010000 MOV #FMT22,(R0)+ ;FORMAT=16 BIT WORDS
6587 ;CYLINDER=0
6588 012722 012720 000001 28: MOV #1,(R0)+ ;TRACK=0, SECTOR=1, KEYS=0
6589 012726 005020 CLR (R0)+ ;KEY1=0

```

```

6590 012730 005020          CLR      (R0)+      ;KEY2=0
6591 012732 012705 000400  MOV      @256,,R5   ;COUNTER
6592 012736 012720 000000 36:  MOV      @0,(R0)+  ;MOVE ALL ZEROS FOR DATA
6593 012742 005305          DEC      R5
6594 012744 001374          BNE     36         ;BRANCH IF DATA NOT COMPLETE
6595 012746 012777 000001 166662 MOV      @1,@RHDST ;TRACK=0 SECTOR=1
6596
6597
(1)
(1) 012754 004767 011544     JSR     PC,CHECKT   ;CHECK DVA, RDY, DPR, DRY
(1)
6598
6599 012760 013711 002042     MOV     @WRIFOR,@R1 ;GET READY FOR WRITE HEADER AND
6600                                ;DATA WITH 62 IN RHC61
6601 012764 005037 001774     CLR     @ERFLG6     ;CLEAR ERROR FLAG
6602 012770 012777 010000 166644 MOV     @FMT22,@RHOF ;FORMAT BIT=1 (16 BIT WORDS)
6603 012776 005077 166642     CLR     @RHCA       ;CYLINDER =0
6604 013002 004737 033776     JSR     PC,@COMWHD  ;WRITE HEADER AND DATA
6605
6606                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6607                                ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
6608                                ;IS GOOD IE, ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
6609                                ;ALL ZEROS AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
6610                                ;THEY ARE ALL ZEROS
6611
6612 013006 005737 001774     TST     @ERFLG6     ;HAS ANY ERRORS OCCURED?
6613                                ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
6614
(1) 013012 001053          BNE     TST17      ;;BRANCH IF YES
(1)
6615
6616
6617                                ;COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
6618 013014 023737 027126 033734 CMP     @GECC1,@WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
6619 013022 001402          BEQ     66         ;BRANCH IF GOOD
6620 013024 104031          ERROR   31        ;LOW ORDER ECC IN ERROR
6621 013026 000405          BR      78         ;BRANCH TO CONTINUE
6622 013030 023737 027130 033736 68:  CMP     @GECC2,@WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
6623 013036 001401          BEQ     78         ;BRANCH IF GOOD
6624 013040 104031          ERROR   31        ;HIGH ORDER ECC IN ERROR
6625
6626
6627 013042 004737 024706     78:  JSR     PC,@CHECKE  ;CHECK DVA,RDY,DRY,DPR
6628
6629
6630
6631                                ;FILL "REINTO" BUFFER WITH EXPECTED DATA
6632 013046 004037 024406     JSR     R0,@CLAREA  ;FILL REINTO BUFFER
6633 013052 003130          REINTO  ;FROM
6634 013054 004126          REINTO+<255,*2>  ;TO
6635 013056 000000          ,WORD  0         ;DATA
6636
6637 013060 013737 027126 004130 MOV     @GECC1,@REINTO+<256,*2>;FILL ECC1
6638 013066 013737 027130 004132 MOV     @GECC2,@REINTO+<257,*2>;FILL ECC2

```

```
6639 013074 004037 024406 JSR R0,0:CLAREA ;FILL REST
6640 013100 004134 REINTO+<250,*2> ;FROM
6641 013102 004170 REINTO+<272,*2> ;TO
6642 013104 000000 0 ;DATA
6643
6644
6645 013106 005037 001774 CLR 00ERFLG8 ;CLEAR ERROR FLAG
6646
6647
6648 ;NOW COMPARE "DISK" BUFFER WITH "REINTO"
6649 013112 004037 025342 JSR R0,0:COMPAR ;CHECK
6650 013116 003130 REINTO ;GOOD BUFFER
6651 013120 032734 DISK ;TEST BUFFER
6652 013122 000402 250, ;NUMBER OF WORDS CHECKED
6653 013124 013132 40 ;RETURN POINT FOR ERROR HEADER
6654 013126 013136 50 ;RETURN POINT FOR ERROR DATA
6655
6655 (1) 013130 013142 TST17 ;RETURN FOR GOOD COMPARISON
6655 (1)
6656 013132 104007 40: ERROR 7 ;READ ERROR 10 NEXT
6657 013134 000207 RTS PC ;RETURN TO COMPARE
6658 013136 104010 50: ERROR 10 ;WORD NOS 1 TO 256 ARE
6659 ;DATA WORDS
6660 ;WORD NOS 257 AND 258
6661 ;ARE ECC WHICH ARE CHECKED
6662 ;WORD NOS 259
6663 ;IS DATA GAP
6664 ;WORD NOS 260 TO 273
6665 ;ARE TOLERANCE GAP
6666 013140 000207 RTS PC ;RETURN TO COMPARE
6667
6668
6669
6670
6671
6682
6683 ;*****
6683 (3) ;*TEST 17 READ ECC ENABLED 1A
6683 (4)
6683 (4) ;* THIS IS AN ECC READ DATA TEST
6683 (4) ;* ERROR CORRECTION IS ENABLED
6683 (4) ;* NO ERROR IS INSERTED
6683 (4) ;* GOOD DATA USED IS 256 WORDS OF 0
6683 (4) ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
6683 (4) ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
6683 (4)
6683 (3) ;*****
6684 (2) 013142 000004 TST17: SCOPE
6684 013144 012706 001000 NOV 0:STACK,SP ;RESET STACK
6685
6686
6686 (1) 013150 012737 000017 004174 NOV 0:TTNO,0:TSTNM ;THIS SAVES TEST NUMBER
6686 (1)
6687
```

```

6688
6689
6690
6691
6692 013156 012746 000000      MOV    00,    -(SP)    ;DATA TO BE READ
6693 013162 012705 000400      MOV    0256,,  R5     ;COUNTER
6694 013166 012700 032734      MOV    0DISK,  R0     ;START OF SIMULATED DISK DATA
6695 013172 011620      101  MOV    (SP),  (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
6696 013174 005305      DEC    R5            ;COUNT
6697 013176 001375      BNE    10           ;BRANCH IF 256 NOT COMPLETE
6698 013200 005726      TST    (SP)+        ;UNDO -(SP)
6699 013202 022020      CMP    (R0)+,(R0)+  ;JUMP OVER THE TWO ECC WORDS
6700 013204 012705 000017      MOV    015,,  R5     ;1 DATA GAP
6701
6702 013210 005020      201  CLR    (R0)+        ;CLEAR DATA GAP, AND
6703 013212 005305      DEC    R5            ;TOLERANCE GAP
6704 013214 001375      BNE    20           ;BRANCH IF NOT COMPLETE
6705
6706
6707 013216 004737 027722      JSR    PC,00FILLEC  ;INSERT THE TWO ECC WORDS ON THE DISK
6708
6709
6710
6711
6712 013222 012737 177777 002012      MOV    0-1,00TSECC  ;THIS IS AN ECC TEST
6713 013230 005037 027140      CLR    00POSITI    ;CLEAR ERROR POSITION COUNTER
6714 013234 013737 027134 027136      MOV    00NCODE,00NCOUNT ;TEMPORARY N-CODE COUNTER
6715 013242 013737 027142 027150      MOV    00HARDER,00HADTMP ;TEMPORARY HARD ERROR COUNTER
6716 013250 005037 027126      CLR    00GECC1     ;ECC LOW ORDER TO BE GENERATED
6717 013254 005037 027130      CLR    00GECC2     ;ECC HIGH ORDER TO BE GENERATED
6718 013260 005037 027144      CLR    00DATENV    ;CLEAR DATA ENVELOPE CLOCK COUNT
6719 013264 005037 027146      CLR    00ZCODE     ;CLEAR LEADING ZEROS CLOCK COUNT
6720
6721
6722
6723
6724 013270 012737 010000 031016      MOV    0FMT22,00CYL ;16 BITS PER WORD
6725
6726 013276 112737 000000 031021      MOV    00,,  00SECOTR+1 ;TRACK 0
6727 013304 112737 000000 031020      MOV    00,,  00SECOTR ;SECTOR 0
6728 013312 012737 000000 031022      MOV    00,,  00KEY1  ;KEY1=0
6729 013320 012737 000000 031024      MOV    00,,  00KEY2  ;KEY2=0
6730 013326 012737 000400 031076      MOV    0256,,  00DAWORD ;NO. OF DATA WORDS
6731 013334 005037 031026      CLR    00X         ;THIS IS A READ COMMAND
6732 013340 004537 025646      JSR    R5,00CRC    ;GO TO CALCULATE CRC
6733 013344 031016      CYL
6734 013346 032716      WCRC
6735
6736
6737
6738
6739
6740 013350 004737 024470      ;THESE ARE REGULAR SETUPS
6741 013354 012777 177374 166242      JSR    PC,00CLDISK ;SETUP GENERAL REGISTERS
        MOV    0-256,-4,,0RHWC ;256, DATA 4 HEADER WORDS

```

6742	013362	012777	003130	166236	MOV	0REINT0,0RHBA	;STARTING ADDRESS OF READ BUFFER
6743	013370	112746	000000		MOVB	00, -(SP)	;IN LOWER BYTE GET SECTOR
6744	013374	112766	000000	000001	MOVB	00, 1(SP)	;GET TRACK IN HIGHER BYTE
6745	013402	012677	166230		MOV	(SP)+, 0RHDS1	;TRACK/SECTOR IN RHDS1
6746	013406	012777	010000	166226	MOV	0FMT22,0RHOF	;16 BITS PER WORD
6747							;ECC CORRECTION NOT INHIBIT
6748							;BECAUSE ECC IS NOT GOING
6749							;TO BE CHECKED
6750	013414	005077	166224		CLR	0RHCA	;CYLINDER 0
6751							
6752	013420	004737	024524		JSR	PC, 00CHECKT	;CHECK FOR DVA,RDY,MOL,DPR,DRY
6753							
6754	013424	013711	002046		MOV	00REFOR,0R1	;READ HEADER AND DATA=72
6755	013430	005037	001774		CLR	00ERFLG0	;CLEAR ERROR FLAG
6756	013434	004737	030706		JSR	PC, 00CONHD	;READ HEADER AND DATA
6757							;IF THERE ARE READ ERRORS THEN
6758							;ECC WILL NOT BE CHECKED
6759							
6760							
6761							
6762							;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6763							;FROM THE "CONHD" ROUTINE THAT MEANS SECTOR GAP,
6764							;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6765							;SYNC BYTE HAVE GONE BY AND SYNC0 WERE CORRECTLY
6766							;DETECTED
6767							;HEADER AND DATA ARE TO BE CHECKED,
6768							;IN CHECKING READ DATA THE WRITE FROM BUFFER
6769							; "NRFROM" IS FILLED WITH EXPECTED DATA AND
6770							;COMPARISONS ARE MADE
6771	013440	005737	001774		TST	00ERFLG0	;ANY ERRORS ALREADY THERE
6772							
(1)	013444	001077			BNE	TST20	;BRANCH IF YES
(1)							
6773	013446	004737	024170		JSR	PC,00PUTREG	;SAVE REGISTERS
6774	013452	005737	001704		TST	00ER1	;NO ERRORS SHOULD BE SET
6775	013456	001401			BEQ	00	;BRANCH IF NO ERRORS SET
6776	013460	104032			ERROR	32	;32 BIT ECC REGISTER SHOULD BE ZERO
6777							;ONLY 11 OF THE 32 BITS CAN BE SEEN
6778							;IN THE PATTERN REGISTER
6779							;DCK SHOULD BE SET IN RHER1
6780	013462	013746	027126	601	MOV	00GECC1,-(SP)	;GET PATTERN REGISTER
6781	013466	042716	174000		BIC	0174000,(SP)	;KEEP ONLY 11 BITS
6782	013472	022637	001734		CMP	(SP)+,00EC2	;COMPARE PATTERN REGISTER
6783	013476	001401			BEQ	70	;BRANCH IF GOOD
6784	013500	104032			ERROR	32	;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
6785							
6786							
6787							
6788							
6789							;ADD 16 MAINTENANCE CLOCKS TO
6790							;BRING EBL DOWN
6791							
6792	013502	012700	000020	701	MOV	016,,R0	;COUNTER
6793	013506	052777	000002	166136 001	BIS	0MCLK,0RHMR	;SET CLOCK

```

6794 013514 042777 000002 166130      BIC      0MCLK,0RHNR      ;CLEAR CLOCK
6795 013522 005300      DEC      R0              ;COUNT
6796 013524 001370      BNE      00              ;BRANCH IF 16 CLOCKS NOT DONE
6797 013526 004737 024706      JSR      PC,00CHECKE     ;CHECK DVA,DRY,RDY,DPR
6798 013532 012700 002064      MOV      0WRFROM,R0     ;GETTING READY TO FILL EXPECTED DATA
6799 013536 012720 010000      MOV      00IFMT22,(R0)+ ;CYLINDER 0
6800 013542 112746 000000      MOV      00, -(SP)      ;IN LOWER BYTE GET SECTOR
6801 013546 112766 000000 000001      MOV      00, 1(SP)      ;GET TRACK IN HIGHER BYTE
6802 013554 012620      MOV      (SP)+, (R0)+    ;GET TRACK/SECTOR IN BUFFER
6803 013556 012720 000000      MOV      00, (R0)+      ;KEY1 IN BUFFER
6804 013562 012720 000000      MOV      00, (R0)+      ;KEY2 IN BUFFER
6805 013566 012701 000400      MOV      0256,, R1      ;DATA WORD COUNTER
6806 013572 012702 000000      MOV      00, R2         ;DATA
6807 013576 010220      MOV      R2, (R0)+      ;DATA INTO BUFFER
6808 013600 005301      DEC      R1              ;COUNT
6809 013602 001375      BNE      30              ;BRANCH IF 256 NOT DONE
6810
6811
6812 013604 005037 001774      CLR      00ERFLG0       ;CLEAR ERROR FLAG
6813 013610 004737 024170      JSR      PC,00PUTREG     ;SAVE REGISTERS
6814
6815
6816
6817      ;NOW READ DATA BUFFER WILL BE CHECKED
6818
6819 013614 004037 025342      JSR      R0,00COMPAR     ;CHECK
6820 013620 002064      WRFROM      ;GOOD BUFFER
6821 013622 003130      REINTO      ;TEST BUFFER
6822 013624 000404      4+256,     ;NUMBER OF WORDS CHECKED
6823 013626 013634      40         ;RETURN POINT FOR ERROR HEADER
6824 013630 013640      50         ;RETURN POINT FOR ERROR DATA
6825
6826 (1) 013632 013644      TST20      ;RETURN FOR GOOD COMPARISON
6827 (1)
6828 013634 104004      401      ERROR 4        ;READ NEXT ERROR
6829 013636 000207      RTS      PC              ;RETURN TO "COMPAR"
6830 013640 104005      501      ERROR 5        ;WORD NOS 1 TO 4 ARE
6831 013642 000207      RTS      PC              ;HEADER WORDS
6832                                     ;5 TO 260 ARE DATA WORDS
6833                                     ;RETURN TO "COMPAR"
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847      ;*****
6848 (3)      ;*TEST 20      READ ECC ENABLED 1B
6849 (4)
6850 (4)      ;* THIS IS AN ECC READ DATA TEST
6851 (4)      ;* ERROR CORRECTION IS ENABLED
6852 (4)      ;* A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32
6853 (4)      ;* GOOD DATA USED IS 256 WORDS OF 0
6854 (4)      ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
6855 (4)      ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA

```

```

(4)
(3)
(2) 013644 000004
6848 013646 012706 001000
6849
6850
(1) 013652 012737 000020 004174
(1)
6851
6852
6853
6854
6855
6856 013660 012746 000000
6857 013664 012705 000400
6858 013670 012700 032734
6859 013674 011620
6860 013676 005305
6861 013700 001375
6862 013702 005726
6863 013704 022020
6864 013706 012705 000017
6865
6866 013712 005020
6867 013714 005305
6868 013716 001375
6869
6870
6871 013720 004737 027722
6872
6873
6874
6875
6876
6877 013724 012737 177777 002012
6878 013732 005037 027140
6879 013736 013737 027134 027136
6880 013744 013737 027142 027150
6881 013752 005037 027126
6882 013756 005037 027130
6883 013762 005037 027144
6884 013766 005037 027146
6885
6886
6887
6888
6889 013772 012737 010000 031016
6890
6891 014000 112737 000000 031021
6892 014006 112737 000000 031020
6893 014014 012737 000000 031022
6894 014022 012737 000000 031024
6895 014030 012737 000400 031076
6896 014036 005037 031026

```

```

;*****
TST20: SCOPE
MOV     #STACK,SP      ;RESET STACK

MOV     #TTNO,#TSTNM  ;THIS SAVES TEST NUMBER

;
;   SETUP FOR WHAT IS TO BE READ
;   HEADER CRC IS RESTORED FROM A SUBROUTINE
;
MOV     #0,      -(SP)  ;DATA TO BE READ
MOV     #256,,   R5     ;COUNTER
MOV     #DISK,   R0     ;START OF SIMULATED DISK DATA
18:    MOV     (SP),  (R0)+ ;MOVE IN DATA ON TO SIMULATED DISK
DEC     R5            ;COUNT
BNE     18          ;BRANCH IF 256 NOT COMPLETE
TST     (SP)+       ;UNDO -(SP)
CMP     (R0)+,(R0)+ ;JUMP OVER THE TWO ECC WORDS
MOV     #15,,    R5    ;1 DATA GAP
;14 TOLERANCE GAP
28:    CLR     (R0)+   ;CLEAR DATA GAP, AND
DEC     R5            ;TOLERANCE GAP
BNE     28          ;BRANCH IF NOT COMPLETE

JSR     PC,#FILLEC    ;INSERT ECC IN PROPER PLACE ON DISK

;THESE ARE FOR ECC TEST ONLY
MOV     #-1,#TSECC    ;THIS IS AN ECC TEST
CLR     #POSITI      ;CLEAR ERROR POSITION COUNTER
MOV     #NCODE,#NCOUNT ;TEMPORARY N-CODE COUNTER
MOV     #HARDER,#HADTMP ;TEMPORARY HARD ERROR COUNTER
CLR     #GECC1       ;ECC LOW ORDER TO BE GENERATED
CLR     #GECC2       ;ECC HIGH ORDER TO BE GENERATED
CLR     #DATENV      ;CLEAR DATA ENVELOPE CLOCK COUNT
CLR     #ZCODE       ;CLEAR LEADING ZEROS CLOCK COUNT

;THESE ARE TO SETUP FOR DISKLESS USE ONLY
MOV     #FMT22,#CYL   ;16 BITS PER WORD
;CYLINDER #, FORMAT 16 BITS
MOVB    #0,          #SECOTR+1 ;TRACK #
MOVB    #0,          #SECOTR ;SECTOR #
MOV     #0,          #KEY1 ;KEY1=#
MOV     #0,          #KEY2 ;KEY2=#
MOV     #256,,      #DAWORD ;NO. OF DATA WORDS
CLR     #0X          ;THIS IS A READ COMMAND

```


6897	014042	004537	025646		JSR	RS,00CRC	;GO TO CALCULATE CRC
6898	014046	031016			CYL		
6899	014050	032716			WCRC		
6900							
6901							
6902							
6903							
6904							
6905							
6906	014052	012737	100000	032736	MOV	0100000,00DISK+2	;FORCE ERROR ON BIT NUMBER 32
6907							;80 ERROR POSITION REGISTER WILL SHOW
6908							;22
6909	014060	012737	000026	014226	NOV	022,,0000	;INSERT POSITION REG.
6910							
6911							
6912							
6913	014066	004737	024470		JSR	PC,00CLDISK	;SETUP GENERAL REGISTERS
6914	014072	012777	177374	165524	NOV	0-256,-4,,0RHWC	;256, DATA 4 HEADER WORDS
6915	014100	012777	003130	165520	NOV	0REINT0,0RHBA	;STARTING ADDRESS OF READ BUFFER
6916	014106	112746	000000		NOVB	00, -(SP)	;IN LOWER BYTE GET SECTOR
6917	014112	112766	000000	000001	NOVB	00, 1(SP)	;GET TRACK IN HIGHER BYTE
6918	014120	012677	165512		NOV	(SP)+, 0RHDST	;TRACK/SECTOR IN RHDST
6919	014124	012777	010000	165510	NOV	0FMT22,0RHOF ;16	BITS PER WORD
6920							;ECC CORRECTION NOT INHIBIT
6921							;BECAUSE ECC IS NOT GOING
6922							;TO BE CHECKED
6923	014132	005077	165506		CLR	0RHCA	;CYLINDER 0
6924							
6925	014136	004737	024524		JSR	PC, 00CHECKT	;CHECK FOR DVA,RDY,NOL,DPR,DRY
6926							
6927	014142	013711	002046		MOV	00REFOR,0R1	;READ HEADER AND DATA=72
6928	014146	005037	001774		CLR	00ERFLG0	;CLEAR ERROR FLAG
6929	014152	004737	030706		JSR	PC, 00COMHD	;READ HEADER AND DATA
6930							;IF THERE ARE READ ERRORS THEN
6931							;ECC WILL NOT BE CHECKED
6932							
6933							
6934							
6935							
6936							
6937							
6938							
6939							
6940							
6941							
6942							
6943							
6944	014156	005737	001774		TST	00ERFLG0	;ANY ERRORS ALREADY THERE
6945							
(1)	014162	001074			BNE	TST21	;BRANCH IF YES
(1)							
6946	014164	004737	024170		JSR	PC,00PUTREG	;SAVE REGISTERS
6947	014170	022737	100000	001704	CHP	0DCK,00ER1	;ONLY DATA CHECK ERROR SHOULD BE SET
6948	014176	001401			BEG	00	;BRANCH IF YES


```

7001 014350 104005          501  ERROR  5          ;WORD NOS 1 TO 4 ARE
7002                                     ;HEADER WORDS
7003                                     ;5 TO 260 ARE DATA WORDS
7004 014352 000207          RTS    PC          ;RETURN TO "COMPAR"
7005
7016
7017          ;*****
(3)          ;TEST 21      READ ECC ENABLED 1C
(4)
(4)          ;*      THIS IS AN ECC READ DATA TEST
(4)          ;*      ERROR CORRECTION IS ENABLED
(4)          ;*      A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 21 THRU 32
(4)          ;*      GOOD DATA USED IS 256 WORDS OF 0
(4)          ;*      COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
(4)          ;*      TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
(4)
(3)          ;*****
(2) 014354 000004          TST21: SCOPE
7018 014356 012706 001000          MOV    0STACK,SP      ;RESET STACK
7019
7020          (1) 014362 012737 000021 004174          MOV    0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
7021
7022          ;      SETUP FOR WHAT IS TO BE READ
7023          ;      HEADER CRC IS RESTORED FROM A SUBROUTINE
7024
7025
7026 014370 012746 000000          MOV    00, -(SP)      ;DATA TO BE READ
7027 014374 012705 000400          MOV    0256,, R5      ;COUNTER
7028 014400 012700 032734          MOV    0DISK, R0      ;START OF SIMULATED DISK DATA
7029 014404 011620          101  MOV    (SP), (R0)+    ;MOVE IN DATA ON TO SIMULATED DISK
7030 014406 005305          DEC    R5              ;COUNT
7031 014410 001375          BNE    10              ;BRANCH IF 256 NOT COMPLETE
7032 014412 005726          TST    (SP)+          ;UNDO -(SP)
7033 014414 022020          CMP    (R0)+,(R0)+    ;JUMP OVER THE TWO ECC WORDS
7034 014416 012705 000017          MOV    015,, R5      ;1 DATA GAP
7035                                     ;14 TOLERANCE GAP
7036 014422 005020          201  CLR    (R0)+          ;CLEAR DATA GAP, AND
7037 014424 005305          DEC    R5              ;TOLERANCE GAP
7038 014426 001375          BNE    20              ;BRANCH IF NOT COMPLETE
7039
7040
7041 014430 004737 027722          JSR    PC,00FILLEC    ;INSERT THE TWO ECC WORDS ON THE DISK
7042                                     ;IN THE CORRECT PLACE
7043
7044          ;THESE ARE FOR ECC TEST ONLY
7045
7046 014434 012737 177777 002012          MOV    0-1,00TSECC    ;THIS IS AN ECC TEST
7047 014442 005037 027140          CLR    00POSITI      ;CLEAR ERROR POSITION COUNTER
7048 014446 013737 027134 027136          MOV    00NCODE,00NCOUNT ;TEMPORARY N-CODE COUNTER
7049 014454 013737 027142 027150          MOV    00HARDER,00HADTMP ;TEMPORARY HARD ERROR COUNTER
7050 014462 005037 027126          CLR    00GECC1       ;ECC LOW ORDER TO BE GENERATED
7051 014466 005037 027130          CLR    00GECC2       ;ECC HIGH ORDER TO BE GENERATED

```

```

7052 014472 005037 027144 CLR 00DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
7053 014476 005037 027146 CLR 00ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
7054
7055
7056 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
7057
7058 014502 012737 010000 031016 NOV 0FMT22,00CYL ;16 BITS PER WORD
7059 ;CYLINDER 0, FORMAT 16 BITS
7060 014510 112737 000000 031021 MOVB 00, 00SECOTR+1 ;TRACK 0
7061 014516 112737 000000 031020 MOVB 00, 00SECOTR ;SECTOR 0
7062 014524 012737 000000 031022 NOV 00, 00KEY1 ;KEY1=0
7063 014532 012737 000000 031024 NOV 00, 00KEY2 ;KEY2=0
7064 014540 012737 000400 031076 NOV 0256,, 00DAWORD ;NO. OF DATA WORDS
7065 014546 005037 031026 CLR 00X ;THIS IS A READ COMMAND
7066 014552 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
7067 014556 031016 CYL
7068 014560 032716 WCRC
7069
7070
7071 ;THIS IS TO INSERT ERROR
7072 ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
7073 ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
7074 ;THIS MOVE
7075 014562 012737 177760 032736 MOV 0177760,00DISK+2 ;FORCE ERROR ON BIT NUMBER 21 THRU 32
7076 ;80 ERROR POSITION REGISTER WILL SHOW
7077 ;22
7078 014570 012737 010040 014736 MOV 04120,,0000 ;INSERT POSITION REG.
7079
7080
7081 ;THESE ARE REGULAR SETUPS
7082 014576 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
7083 014602 012777 177374 165014 MOV 0-256,-4,,00RHWC ;256, DATA 4 HEADER WORDS
7084 014610 012777 003130 165010 MOV 0REINT0,00RHDA ;STARTING ADDRESS OF READ BUFFER
7085 014616 112746 000000 MOVB 00, -(SP) ;IN LOWER BYTE GET SECTOR
7086 014622 112766 000000 000001 MOVB 00, 1(SP) ;GET TRACK IN HIGHER BYTE
7087 014630 012677 165002 NOV (SP)+, 00RHDBT ;TRACK/SECTOR IN RHDBT
7088 014634 012777 010000 165000 MOV 0FMT22,00HOF ;16 BITS PER WORD
7089 ;ECC CORRECTION NOT INHIBIT
7090 ;BECAUSE ECC IS NOT GOING
7091 ;TO BE CHECKED
7092 014642 005077 164776 CLR 00RHCA ;CYLINDER 0
7093
7094 014646 004737 024524 JSR PC, 00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
7095
7096 014652 013711 002046 MOV 00REFOR,00R1 ;READ HEADER AND DATA=72
7097 014656 005037 001774 CLR 00ERFLG ;CLEAR ERROR FLAG
7098 014662 004737 030706 JSR PC, 00CONHD ;READ HEADER AND DATA
7099 ;IF THERE ARE READ ERRORS THEN
7100 ;ECC WILL NOT BE CHECKED
7101
7102
7103 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7104 ;FROM THE "CONHD" ROUTINE THAT MEANS SECTOR GAP,
7105 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND

```

```
7106                                     ;SYNC BYTE HAVE GONE BY AND SYNCB WERE CORRECTLY
7107                                     ;DETECTED
7108                                     ;HEADER AND DATA ARE TO BE CHECKED,
7109                                     ;IN CHECKING READ DATA THE WRITE FROM BUFFER
7110                                     ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
7111                                     ;COMPARISONS ARE MADE
7112
7113 014666 005737 001774                TST      00ERFLG8                ;ANY ERRORS ALREADY THERE
7114
7115 (1) 014672 001103                    BNE      TST22      ;BRANCH IF YES
7116 (1)
7117 014674 004737 024170                JSR      PC,00PUTREG          ;SAVE REGISTERS
7118 014700 022737 100000 001704        CMP      0DCK,00ER1          ;ONLY DATA CHECK ERROR SHOULD BE SET
7119 014706 001401                        BEQ      68                  ;BRANCH IF YES
7120 014710 104032                        ERROR    32                  ;32 BIT ECC REGISTER SHOULD BE NON
7121                                     ;ZERO
7122                                     ;ONLY 11 OF THE 32 BITS CAN BE SEEN
7123                                     ;IN THE PATERN REGISTER
7124 014712 013746 027126 68:            MOV      00GECC1,-(SP)        ;DCK SHOULD BE SET IN RHER1
7125 014716 042716 174000                BIC      0174000,(SP)        ;GET PATTERN REGISTER
7126 014722 022637 001734                CMP      (SP)+,00EC2        ;KEEP ONLY 11 BITS
7127 014726 001401                        BEQ      78                  ;COMPARE PATTERN REGISTER
7128 014730 104032                        ERROR    32                  ;BRANCH IF GOOD
7129                                     ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
7130 014732 004037 027550 78:            JSR      R0,00ECORR          ;GO TO ECC CORRECTION PROCESS
7131 014736 000000 88:                    ,WORD                        ;EXPECTED POSITION REG, WHEN CORRECTION
7132                                     ;IS COMPLETE
7133
7134
7135 014740 004737 024170                JSR      PC,00PUTREG          ;SAVE REGISTERS
7136 014744 022737 100100 001704        CMP      0DCK|ECH,00ER1     ;WITH ERRORS INSERTED IN BIT POSITION 21
7137                                     ;THRU 32 HARD ERROR BIT SHOULD SET
7138 014752 001401                        BEQ      98                  ;BRANCH IF GOOD
7139 014754 104036                        ERROR    36                  ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
7140                                     ;32 ECH SHOULD SET
7141
7142
7143
7144 014756 004737 024706 98:            JSR      PC,00CHECKE        ;CHECK DVA,DRY,RDY,DPR
7145 014762 012700 002064                MOV      0WRFROM,R0         ;GETTING READY TO FILL EXPECTED DATA
7146 014766 012720 010000                MOV      00|FMT22,(R0)+     ;CYLINDER 0
7147 014772 112746 000000                MOV      00,-(SP)          ;IN LOWER BYTE GET SECTOR
7148 014776 112766 000000 000001        MOV      00,1(SP)          ;GET TRACK IN HIGHER BYTE
7149 015004 012620                        MOV      (SP)+,(R0)+        ;GET TRACK/SECTOR IN BUFFER
7150 015006 012720 000000                MOV      00,(R0)+          ;KEY1 IN BUFFER
7151 015012 012720 000000                MOV      00,(R0)+          ;KEY2 IN BUFFER
7152 015016 012701 000400                MOV      0256,,R1          ;DATA WORD COUNTER
7153 015022 012702 000000                MOV      00,R2             ;DATA
7154 015026 010220 38:                    MOV      R2,(R0)+          ;DATA INTO BUFFER
7155 015030 005301                        DEC      R1                 ;COUNT
7156 015032 001375                        BNE      38                 ;BRANCH IF 256 NOT DONE
7157
```

```
7158 ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'  
7159 ;NOW THE INSERTED ERROR WILL BE PUT IN  
7160 015034 012737 177760 002076 MOV 0177760,00WRFROM+<5*2> ;INSERTED ERROR  
7161  
7162  
7163  
7164 015042 005037 001774 CLR 00ERFLG8 ;CLEAR ERROR FLAG  
7165 015046 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS  
7166  
7167  
7168 ;NOW READ DATA BUFFER WILL BE CHECKED  
7169  
7170 015052 004037 025342 JSR R0,00COMPAR ;CHECK  
7171 015056 002064 WRFROM ;GOOD BUFFER  
7172 015060 003130 REINTO ;TEST BUFFER  
7173 015062 000404 4+256, ;NUMBER OF WORDS CHECKED  
7174 015064 015072 48 ;RETURN POINT FOR ERROR HEADER  
7175 015066 015076 58 ;RETURN POINT FOR ERROR DATA  
7176  
7176 (1) 015070 015102 TST22 ;RETURN FOR GOOD COMPARISON  
7176 (1)  
7177 015072 104004 481 ERROR 4 ;READ NEXT ERROR  
7178 015074 000207 RTS PC ;RETURN TO "COMPAR"  
7179 015076 104005 581 ERROR 5 ;WORD NOS 1 TO 4 ARE  
7180 ;HEADER WORDS  
7181 ;5 TO 260 ARE DATA WORDS  
7182 015100 000207 RTS PC ;RETURN TO "COMPAR"  
7183  
7184  
7185  
7186  
7187  
7188  
7189  
7190 ;*****  
7190 (3) ;*TEST 22 WRITE ECC TEST 2  
7190 (4)  
7190 (4) ;* THIS IS A WRITE ECC TEST  
7190 (4) ;* WRITE CYLINDER 0, FORMAT 16 BITS PER WORD  
7190 (4) ;* TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256  
7190 (4) ;* OF ALL ONES,  
7190 (4)  
7190 (3) ;*****  
7199 015102 000004 TST22: SCOPE  
7199 015104 012706 001000 MOV #STACK,SP ;RESET STACK  
7200  
7201  
7201 (1) 015110 012737 000022 004174 MOV #TTNO,#TSTNM ;THIS SAVES TEST NUMBER  
7201 (1)  
7202 015116 012700 032636 MOV #SECGAP,R0 ;POINTER  
7203 015122 012701 000460 MOV #304,,R1 ;COUNTER  
7204 015126 005020 181 CLR (R0)+ ;CLEAR SIMULATED DISK AREA  
7205 015130 005301 DEC R1  
7206 015132 001375 BNE 18
```

```

7207 015134 004767 007330 JSR PC,CLDISK ;THIS IS USED TO SET GENERAL REGISTERS
7208
7209 ;THESE ARE FOR ECC TEST ONLY
7210
7211 015140 012737 177777 002012 MOV 0-1,00TSECC ;THIS IS AN ECC TEST
7212 015146 005037 027140 CLR 00POSITI ;CLEAR ERROR POSITION COUNTER
7213 015152 013737 027134 027136 MOV 00NCODE,00NCOUNT ;TEMPORARY N-CODE COUNTER
7214 015160 013737 027142 027150 MOV 00HARDER,00HADTMP ;TEMPORARY HARD ERROR COUNTER
7215 015166 005037 027126 CLR 00GECC1 ;ECC LOW ORDER TO BE GENERATED
7216 015172 005037 027130 CLR 00GECC2 ;ECC HIGH ORDER TO BE GENERATED
7217 015176 005037 027144 CLR 00DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
7218 015202 005037 027146 CLR 00ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
7219
7220
7221
7222
7223 ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
7224
7225 015206 012737 010000 034122 MOV 0FMT22,00WCYL ;FORMAT22=16BIT WORDS AND
7226 ;CYLINDER 0
7227 015214 012737 000001 034124 MOV 01,00WSECTR ;TRACK=0, SECTOR=1
7228 015222 005037 034126 CLR 00WKEY1 ;KEY1=0
7229 015226 005037 034130 CLR 00WKEY2 ;KEY2=0
7230 015232 012737 000400 034162 MOV 0256,,00FNWORD ;256 DATA WORDS
7231 015240 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
7232 015244 034122 WCYL
7233 015246 034132 GCRC
7234
7235 ;THESE ARE REGULAR SETUPS
7236
7237 015250 012777 177374 164346 MOV 0-260,,0RHWC ;256 DATA WORDS 4 HEADER WORDS
7238 015256 012700 002064 MOV 0WRFROM,R0 ;THESE TWO INSTRUCTIONS GETS
7239 015262 010077 164340 MOV R0,0RHDA ;ADDR. OF WRFROM INTO R0 AND
7240 ;BUS ADDRESS REGISTER
7241 015266 012720 010000 MOV 0FMT22,(R0)+ ;FORMAT=16 BIT WORDS
7242 ;CYLINDER=0
7243 015272 012720 000001 201 MOV 01,(R0)+ ;TRACK=0, SECTOR=1, KEYS=0
7244 015276 005020 CLR (R0)+ ;KEY1=0
7245 015300 005020 CLR (R0)+ ;KEY2=0
7246 015302 012705 000400 MOV 0256,,R5 ;COUNTER
7247 015306 012720 177777 301 MOV 0-1,(R0)+ ;MOVE ALL ONES FOR DATA
7248 015312 005305 DEC R5
7249 015314 001374 BNE 30 ;BRANCH IF DATA NOT COMPLETE
7250 015316 012777 000001 164312 MOV 01,0RHDBT ;TRACK=0 SECTOR=1
7251
7252
(1)
(1) 015324 004767 007174 JSR PC,CHECKT ;CHECK DVA, RDY, DPR, DRY
(1)
7253
7254 015330 013711 002042 MOV 00WRIFOR,0R1 ;GET READY FOR WRITE HEADER AND
7255 ;DATA WITH 02 IN RHCS1
7256 015334 005037 001774 CLR 00ERFLG6 ;CLEAR ERROR FLAG
7257 015340 012777 010000 164274 MOV 0FMT22,0RHOF ;FORMAT BIT=1 (16 BIT WORDS)

```

```

7258 015346 005077 164272 CLR BRHCA ;CYLINDER =0
7259 015352 004737 033776 JSR PC,00COMWHD ;WRITE HEADER AND DATA
7260
7261 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7262 ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
7263 ;IS GOOD IE, ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
7264 ;ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
7265 ;THEY ARE ALL ZEROS
7266
7267 015356 005737 001774 TST 00ERFLG0 ;HAS ANY ERRORS OCCURED?
7268 ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
7269
(1) 015362 001053 BNE TST23 ;;BRANCH IF YES
(1)
7270
7271
7272 ;COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
7273 015364 023737 027126 033734 CMP 00GECC1,00NECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
7274 015372 001402 BEQ 60 ;BRANCH IF GOOD
7275 015374 104031 ERROR 31 ;LOW ORDER ECC IN ERROR
7276 015376 000405 BR 70 ;BRANCH TO CONTINUE
7277 015400 023737 027130 033736 601 CMP 00GECC2,00NECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
7278 015406 001401 BEQ 70 ;BRANCH IF GOOD
7279 015410 104031 ERROR 31 ;HIGH ORDER ECC IN ERROR
7280
7281
7282 015412 004737 024706 701 JSR PC,00CHECKE ;CHECK DVA,RDY,DRY,DPR
7283
7284
7285
7286
7287 ;FILL "REINTO" BUFFER WITH EXPECTED DATA
7288 015416 004037 024406 JSR R0,00CLAREA ;FILL REINTO BUFFER
7289 015422 003130 REINTO ;FROM
7290 015424 004126 REINTO+<255,*2> ;TO
7291 015426 177777 ,WORD -1 ;DATA
7292
7293 015430 013737 027126 004130 MOV 00GECC1,00REINTO+<256,*2>;FILL ECC1
7294 015436 013737 027130 004132 MOV 00GECC2,00REINTO+<257,*2>;FILL ECC2
7295 015444 004037 024406 JSR R0,00CLAREA ;FILL REST
7296 015450 004134 REINTO+<258,*2> ;FROM
7297 015452 004170 REINTO+<272,*2> ;TO
7298 015454 000000 0 ;DATA
7299
7300
7301 015456 005037 001774 CLR 00ERFLG0 ;CLEAR ERROR FLAG
7302
7303
7304 ;NOW COMPARE "DISK" BUFFER WITH "REINTO"
7305 015462 004037 025342 JSR R0,00COMPAR ;CHECK
7306 015466 003130 REINTO ;GOOD BUFFER
7307 015470 032734 DISK ;TEST BUFFER
7308 015472 000402 250, ;NUMBER OF WORDS CHECKED
7309 015474 015502 40 ;RETURN POINT FOR ERROR HEADER

```



```

7310 015476 015506          50          ;RETURN POINT FOR ERROR DATA
7311          (1) 015500 015512          TST23          ;RETURN FOR GOOD COMPARISON
          (1)
7312 015502 104007          401      ERROR    7          ;READ ERROR 10 NEXT
7313 015504 000207                   RTS      PC          ;RETURN TO COMPARE
7314 015506 104010          501      ERROR    10         ;WORD NOS 1 TO 256 ARE
7315          ;DATA WORDS
7316          ;WORD NOS 257 AND 258
7317          ;ARE ECC WHICH ARE CHECKED
7318          ;WORD NOS 259
7319          ;IS DATA GAP
7320          ;WORD NOS 260 TO 273
7321          ;ARE TOLERANCE GAP
7322 015510 000207                   RTS      PC          ;RETURN TO COMPARE
7323
7324
7325
7326
7327
7328
7329
7330
7331          ;*****
7332          ;TEST 23          READ ECC ENABLED 2A
7333
7334          ;*          THIS IS AN ECC READ DATA TEST
7335          ;*          ERROR CORRECTION IS ENABLED
7336          ;*          NO ERROR IS INSERTED
7337          ;*          GOOD DATA USED IS 256 WORDS OF 177777
7338          ;*          COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
7339          ;*          TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
7340          ;*****
7341          TST23: SCOPE
7342          MOV          @STACK,SP          ;RESET STACK
7343
7344          (1) 015512 000004
7345          015514 012706 001000
7346
7347          (1) 015520 012737 000023 004174          MOV          @TTNO,@TSTNM          ;THIS SAVES TEST NUMBER
7348          (1)
7349          ;          SETUP FOR WHAT IS TO BE READ
7350          ;          HEADER CRC IS RESTORED FROM A SUBROUTINE
7351
7352          MOV          @-1,          -(SP)          ;DATA TO BE READ
7353          MOV          @256,, R5          ;COUNTER
7354          MOV          @DISK, R0          ;START OF SIMULATED DISK DATA
7355          MOV          (SP), (R0)+          ;MOVE IN DATA ON TO SIMULATED DISK
7356          DEC          R5          ;COUNT
7357          BNE          IS          ;BRANCH IF 256 NOT COMPLETE
7358          TST          (SP)+          ;UNDO -(SP)
7359          CMP          (R0)+,(R0)+          ;JUMP OVER THE TWO ECC WORDS
7360          MOV          @15,, R5          ;1 DATA GAP
7361          ;16 TOLERANCE GAP
7362          CLR          (R0)+          ;CLEAR DATA GAP, AND

```

```

7359 015562 005305          DEC      R5          ;TOLERANCE GAP
7360 015564 001375          BNE      26          ;BRANCH IF NOT COMPLETE
7361
7362
7363 015566 004737 027722    JSR      PC,00FILLEC ;INSERT THE TWO ECC WORDS ON THE DISK
7364                                ;IN THE CORRECT PLACE
7365
7366                                ;THESE ARE FOR ECC TEST ONLY
7367
7368 015572 012737 177777 002012  MOV      0-1,00TSECC          ;THIS IS AN ECC TEST
7369 015600 005037 027140          CLR      00POSITI          ;CLEAR ERROR POSITION COUNTER
7370 015604 013737 027134 027136  MOV      00NCODE,00NCOUNT      ;TEMPORARY N-CODE COUNTER
7371 015612 013737 027142 027150  MOV      00HARDER,00HADTHP      ;TEMPORARY HARD ERROR COUNTER
7372 015620 005037 027126          CLR      00GECC1          ;ECC LOW ORDER TO BE GENERATED
7373 015624 005037 027130          CLR      00GECC2          ;ECC HIGH ORDER TO BE GENERATED
7374 015630 005037 027144          CLR      00DATENV          ;CLEAR DATA ENVELOPE CLOCK COUNT
7375 015634 005037 027146          CLR      00ZCODE          ;CLEAR LEADING ZEROS CLOCK COUNT
7376
7377
7378                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
7379
7380 015640 012737 010000 031016  MOV      0FMT22,00CYL          ;16 BITS PER WORD
7381                                ;CYLINDER 0, FORMAT 16 BITS
7382 015646 112737 000000 031021  MOV      00, 00SECOTR+1 ;TRACK 0
7383 015654 112737 000000 031020  MOV      00, 00SECOTR ;SECTOR 0
7384 015662 012737 000000 031022  MOV      00, 00KEY1 ;KEY1=0
7385 015670 012737 000000 031024  MOV      00, 00KEY2 ;KEY2=0
7386 015676 012737 000400 031076  MOV      0256,, 00DANORD ;NO. OF DATA WORDS
7387 015704 005037 031026          CLR      00X          ;THIS IS A READ COMMAND
7388 015710 004537 025646          JSR      R5,00CRC          ;GO TO CALCULATE CRC
7389 015714 031016          CYL
7390 015716 032716          WCRC
7391
7392
7393
7394
7395                                ;THESE ARE REGULAR SETUPS
7396 015720 004737 024470          JSR      PC,00CLDISK          ;SETUP GENERAL REGISTERS
7397 015724 012777 177374 163672  MOV      0-256,-4,,00RHWC ;256, DATA 4 HEADER WORDS
7398 015732 012777 003130 163666  MOV      00REINTO,00RHBA ;STARTING ADDRESS OF READ BUFFER
7399 015740 112746 000000          MOV      00, -(SP)          ;IN LOWER BYTE GET SECTOR
7400 015744 112766 000000 000001  MOV      00, 1(SP)          ;GET TRACK IN HIGHER BYTE
7401 015752 012677 163660          MOV      (SP)+, 00RNDST ;TRACK/SECTOR IN RNDST
7402 015756 012777 010000 163656  MOV      0FMT22,00%OF ;16 BITS PER WORD
7403                                ;ECC CORRECTION NOT INHIBIT
7404                                ;BECAUSE ECC IS NOT GOING
7405                                ;TO BE CHECKED
7406 015764 005077 163654          CLR      00RHCA          ;CYLINDER 0
7407
7408 015770 004737 024524          JSR      PC, 00CHECKT ;CHECK FOR DVA,RDY,NOL,DPR,DRY
7409
7410 015774 013711 002046          MOV      00REFOR,00R1          ;READ HEADER AND DATA=72
7411 016000 005037 001774          CLR      00ERFLG0          ;CLEAR ERROR FLAG
7412 016004 004737 030706          JSR      PC, 00COMHD ;READ HEADER AND DATA

```

```
7413                                     ;IF THERE ARE READ ERRORS THEN
7414                                     ;ECC WILL NOT BE CHECKED
7415
7416
7417                                     ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7418                                     ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
7419                                     ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
7420                                     ;SYNC BYTE HAVE GONE BY AND SYNCB WERE CORRECTLY
7421                                     ;DETECTED
7422                                     ;HEADER AND DATA ARE TO BE CHECKED,
7423                                     ;IN CHECKING READ DATA THE WRITE FROM BUFFER
7424                                     ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
7425                                     ;COMPARISONS ARE MADE
7426
7427 016010 005737 001774                TST      00ERFLG0                ;ANY ERRORS ALREADY THERE
7428
7429 (1) 016014 001077                    BNE      TST24                ;BRANCH IF YES
7430 (1)
7431 016016 004737 024170                JSR      PC,00PUTREG          ;SAVE REGISTERS
7432 016022 005737 001704                TST      00ER1                ;NO ERRORS SHOULD BE SET
7433 016026 001401                        BEQ      60                    ;BRANCH IF NO ERRORS SET
7434 016030 104032                        ERROR   32                    ;32 BIT ECC REGISTER SHOULD BE ZERO
7435                                     ;ONLY 11 OF THE 32 BITS CAN BE SEEN
7436                                     ;IN THE PATTERN REGISTER
7437 016032 013746 027126                NOV      00GECC1,-(SP)        ;DCK SHOULD BE SET IN RHER1
7438 016036 042716 174000                BIC      0174000,(SP)        ;GET PATTERN REGISTER
7439 016042 022637 001734                CNP      (SP)+,00EC2         ;KEEP ONLY 11 BITS
7440 016046 001401                        BEQ      70                    ;COMPARE PATTERN REGISTER
7441 016050 104032                        ERROR   32                    ;BRANCH IF GOOD
7442                                     ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
7443
7444
7445                                     ;ADD 16 MAINTENANCE CLOCKS TO
7446                                     ;BRING EBL DOWN
7447
7448 016052 012700 000020                NOV      016,,R0              ;COUNTER
7449 016056 052777 000002 163566 001    BIS      0MCLK,0RHMR          ;SET CLOCK
7450 016064 042777 000002 163560        BIC      0MCLK,0RHMR          ;CLEAR CLOCK
7451 016072 005300                        DEC      R0                    ;COUNT
7452 016074 001370                        BNE      00                    ;BRANCH IF 16 CLOCKS NOT DONE
7453 016076 004737 024706                JSR      PC,00CHECKE         ;CHECK DVA,DRY,RDY,DPR
7454 016102 012700 002064                NOV      0WRFROM,R0          ;CHECKING READY TO FILL EXPECTED DATA
7455 016106 012720 010000                NOV      00IFMT22,(R0)+      ;GETTING READY TO FILL EXPECTED DATA
7456 016112 112746 000000                NOV      00, -(SP)           ;CYLINDER 0
7457 016116 112766 000000 000001        NOV      00, 1(SP)           ;IN LOWER BYTE GET SECTOR
7458 016124 012620                        NOV      (SP)+, (R0)+        ;GET TRACK IN HIGHER BYTE
7459 016126 012720 000000                NOV      00, (R0)+          ;GET TRACK/SECTOR IN BUFFER
7460 016132 012720 000000                NOV      00, (R0)+          ;KEY1 IN BUFFER
7461 016136 012701 000400                NOV      00, (R0)+          ;KEY2 IN BUFFER
7462 016142 012702 177777                NOV      0256,, R1          ;DATA WORD COUNTER
7463 016146 010220                        NOV      0-1, R2            ;DATA
7464 016150 005301                        NOV      R2, (R0)+          ;DATA INTO BUFFER
7465                                     DEC      R1                    ;COUNT
```

```
7465 016152 001375          BNE      30          ;BRANCH IF 256 NOT DONE
7466
7467
7468
7469
7470 016154 005037 001774    CLR      00ERFLG8      ;CLEAR ERROR FLAG
7471
7472 016160 004737 024170    JSR      PC,00PUTREG   ;SAVE REGISTERS
7473
7474                          ;NOW READ DATA BUFFER WILL BE CHECKED
7475
7476 016164 004037 025342    JSR      R0,00COMPAR   ;CHECK
7477 016170 002064            WRFROM                ;GOOD BUFFER
7478 016172 003130            REINTO                ;TEST BUFFER
7479 016174 000404            4+256,                ;NUMBER OF WORDS CHECKED
7480 016176 016204            40                    ;RETURN POINT FOR ERROR HEADER
7481 016200 016210            50                    ;RETURN POINT FOR ERROR DATA
7482
7483 (1) 016202 016214          TST24                ;RETURN FOR GOOD COMPARISON
7484 (1)
7485 016204 104004          401      ERROR      4          ;READ NEXT ERROR
7486 016206 000207          501      RTS        PC          ;RETURN TO "COMPAR"
7487 016210 104008          501      ERROR      5          ;WORD NOS 1 TO 4 ARE
7488                                ;HEADER WORDS
7489                                ;5 TO 260 ARE DATA WORDS
7490                                ;RETURN TO "COMPAR"
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504 ;*****
7505 ;*TEST 24      READ ECC ENABLED 2B
7506
7507 ;*      THIS IS AN ECC READ DATA TEST
7508 ;*      ERROR CORRECTION IS ENABLED
7509 ;*      A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32
7510 ;*      GOOD DATA USED IS 256 WORDS OF 177777
7511 ;*      COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
7512 ;*      TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
7513 ;*****
7514 (2) 016214 000004          TST241  SCOPE
7515 016216 012706 001000    NOV      0STACK,SP      ;RESET STACK
7516
7517
7518 (1) 016222 012737 000024 004174    NOV      0TTNO,00TSTNM  ;THIS SAVES TEST NUMBER
7519 (1)
7520
7521
7522
7523 ;      SETUP FOR WHAT IS TO BE READ
7524 ;      HEADER CRC IS RESTORED FROM A SUBROUTINE
7525
7526 016230 012746 177777    NOV      0-1,  -(SP)    ;DATA TO BE READ
```

```

7514 016234 012705 000400      MOV      #256,, R5      ;COUNTER
7515 016240 012700 032734      MOV      @DISK, R0     ;START OF SIMULATED DISK DATA
7516 016244 011620          18:  MOV      (SP), (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
7517 016246 005305          DEC      R5            ;COUNT
7518 016250 001375          BNE     18            ;BRANCH IF 256 NOT COMPLETE
7519 016252 005726          TST     (SP)+         ;UNDO -(SP)
7520 016254 022020          CMP     (R0)+,(R0)+   ;JUMP OVER THE TWO ECC WORDS
7521 016256 012705 000017      MOV      #15,, R5     ;1 DATA GAP
7522                          ;14 TOLERANCE GAP
7523 016262 005020          28:  CLR     (R0)+         ;CLEAR DATA GAP, AND
7524 016264 005305          DEC     R5            ;TOLERANCE GAP
7525 016266 001375          BNE     28            ;BRANCH IF NOT COMPLETE
7526
7527
7528 016270 004737 027722      JSR     PC,@FILLEC   ;INSERT ECC IN PROPER PLACE ON DISK
7529
7530
7531
7532                          ;THESE ARE FOR ECC TEST ONLY
7533
7534 016274 012737 177777 002012  MOV      #0-1,@TSECC   ;THIS IS AN ECC TEST
7535 016302 005037 027140      CLR     @POSITI      ;CLEAR ERROR POSITION COUNTER
7536 016306 013737 027134 027136  MOV      @NCODE,@NCOUNT ;TEMPORARY N-CODE COUNTER
7537 016314 013737 027142 027150  MOV      @HARDER,@HADTMP ;TEMPORARY HARD ERROR COUNTER
7538 016322 005037 027126      CLR     @GECC1       ;ECC LOW ORDER TO BE GENERATED
7539 016326 005037 027130      CLR     @GECC2       ;ECC HIGH ORDER TO BE GENERATED
7540 016332 005037 027144      CLR     @DATENV      ;CLEAR DATA ENVELOPE CLOCK COUNT
7541 016336 005037 027146      CLR     @ZCODE       ;CLEAR LEADING ZEROS CLOCK COUNT
7542
7543
7544                          ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
7545
7546 016342 012737 010000 031016  MOV      @FMT22,@CYL   ;16 BITS PER WORD
7547                          ;CYLINDER 0, FORMAT 16 BITS
7548 016350 112737 000000 031021  MOVB    #0, @SECOTR+1 ;TRACK 0
7549 016356 112737 000000 031020  MOVB    #0, @SECOTR  ;SECTOR 0
7550 016364 012737 000000 031022  MOV     #0, @KEY1    ;KEY1=0
7551 016372 012737 000000 031024  MOV     #0, @KEY2    ;KEY2=0
7552 016400 012737 000400 031076  MOV     #256,, @DANORD ;NO. OF DATA WORDS
7553 016406 005037 031026      CLR     #0X          ;THIS IS A READ COMMAND
7554 016412 004537 025646      JSR     R5,@CRC      ;GO TO CALCULATE CRC
7555 016416 031016      CYL
7556 016420 032716      WCRC
7557
7558
7559
7560                          ;THIS IS TO INSERT ERROR
7561                          ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
7562                          ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
7563                          ;THIS MOVE
7564 016422 012737 077777 032736  MOV     #077777,@DISK+2 ;FORCE ERROR ON BIT NUMBER 32
7565                          ;80 ERROR POSITION REGISTER WILL SHOW
7566 016430 012737 000026 016576  MOV     #22,,@ERR     ;INSERT POSITION REG,
7567

```

```

7568
7569 ;THESE ARE REGULAR SETUPS
7570 016436 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
7571 016442 012777 177374 163154 MOV #=256,-4,,0RHWC ;256, DATA 4 HEADER WORDS
7572 016450 012777 003130 163150 MOV 0REINT0,0RHBA ;STARTING ADDRESS OF READ BUFFER
7573 016456 112746 000000 MOV# 00, -(SP) ;IN LOWER BYTE GET SECTOR
7574 016462 112766 000000 000001 MOV# 00, 1(SP) ;GET TRACK IN HIGHER BYTE
7575 016470 012677 163142 MOV (SP)+, 0RHDS1 ;TRACK/SECTOR IN RHDS1
7576 016474 012777 010000 163140 MOV 0FMT22,0RHOF ;16 BITS PER WORD
7577 ;ECC CORRECTION NOT INHIBIT
7578 ;BECAUSE ECC IS NOT GOING
7579 ;TO BE CHECKED
7580 016502 005077 163136 CLR 0RHCA ;CYLINDER 0
7581
7582 016506 004737 024524 JSR PC, 00CHECKT ;CHECK FOR DVA, RDY, MOL, DPR, DRY
7583
7584 016512 013711 002046 MOV 00REFOR,0R1 ;READ HEADER AND DATA=72
7585 016516 005037 001774 CLR 00ERFLG0 ;CLEAR ERROR FLAG
7586 016522 004737 030706 JSR PC, 00CONHD ;READ HEADER AND DATA
7587 ;IF THERE ARE READ ERRORS THEN
7588 ;ECC WILL NOT BE CHECKED
7589
7590
7591 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7592 ;FROM THE "CONHD" ROUTINE THAT MEANS SECTOR GAP,
7593 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
7594 ;SYNC BYTE HAVE GONE BY AND SYNC0 WERE CORRECTLY
7595 ;DETECTED
7596 ;HEADER AND DATA ARE TO BE CHECKED,
7597 ;IN CHECKING READ DATA THE WRITE FROM BUFFER
7598 ;"WRFRON" IS FILLED WITH EXPECTED DATA AND
7599 ;COMPARISONS ARE MADE
7600
7601 016526 005737 001774 TST 00ERFLG0 ;ANY ERRORS ALREADY THERE
7602
7603 (1) 016532 001074 BNE TST25 ;BRANCH IF YES
7604 (1)
7605 016534 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
7606 016540 022737 100000 001704 CMP 0DCK,00ER1 ;ONLY DATA CHECK ERROR SHOULD BE SET
7607 016546 001401 BEQ 60 ;BRANCH IF YES
7608 016550 104032 ERROR 32 ;32 BIT ECC REGISTER SHOULD BE NON
7609 ;ZERO
7610 ;ONLY 11 OF THE 32 BITS CAN BE SEEN
7611 016552 013746 027126 601 MOV 00GECC1,-(SP) ;GET PATTERN REGISTER
7612 016556 042716 174000 BIC 0174000,(SP) ;KEEP ONLY 11 BITS
7613 016562 022637 001734 CMP (SP)+,00EC2 ;COMPARE PATTERN REGISTER
7614 016566 001401 BEQ 70 ;BRANCH IF GOOD
7615 016570 104032 ERROR 32 ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
7616
7617 016572 004037 027550 701 JSR R0,00ECORR ;GO TO ECC CORRECTION PROCESS
7618 016576 000026 001 22, ;EXPECTED POSITION REG, WHEN CORRECTION
7619 ;IS COMPLETE

```

```

7620
7621
7622
7623 016600 004737 024706 JSR PC,00CHECKE ;CHECK DVA,DRY,RDY,DPR
7624 016604 012700 002064 MOV 0WRFRON,R0 ;GETTING READY TO FILL EXPECTED DATA
7625 016610 012720 010000 MOV 00IFMT22,(R0)+ ;CYLINDER 0
7626 016614 112746 000000 MOVB 00, -(SP) ;IN LOWER BYTE GET SECTOR
7627 016620 112706 000000 000001 MOVB 00, 1(SP) ;GET TRACK IN HIGHER BYTE
7628 016626 012620 000000 MOV (SP)+, (R0)+ ;GET TRACK/SECTOR IN BUFFER
7629 016630 012720 000000 MOV 00, (R0)+ ;KEY1 IN BUFFER
7630 016634 012720 000000 MOV 00, (R0)+ ;KEY2 IN BUFFER
7631 016640 012701 000400 MOV 0256,, R1 ;DATA WORD COUNTER
7632 016644 012702 177777 MOV 0-1, R2 ;DATA
7633 016650 010220 301 MOV R2, (R0)+ ;DATA INTO BUFFER
7634 016652 005301 DEC R1 ;COUNT
7635 016654 001375 BNE 30 ;BRANCH IF 256 NOT DONE
7636
7637 ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFRON'
7638 ;NOW THE INSERTED ERROR WILL BE PUT IN
7639 016656 012737 077777 002076 MOV 077777,00WRFRON+<5+2> ;INSERTED ERROR
7640
7641
7642 016664 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
7643
7644 016670 005037 001774 CLR 00ERFLG ;CLEAR ERROR FLAG
7645
7646
7647 ;NOW READ DATA BUFFER WILL BE CHECKED
7648
7649 016674 004037 025342 JSR R0,00COMPAR ;CHECK
7650 016700 002064 WRFRON ;GOOD BUFFER
7651 016702 003130 REINTO ;TEST BUFFER
7652 016704 000404 4+256. ;NUMBER OF WORDS CHECKED
7653 016706 016714 40 ;RETURN POINT FOR ERROR HEADER
7654 016710 016720 50 ;RETURN POINT FOR ERROR DATA
7655
(1) 016712 016724 TST25 ;RETURN FOR GOOD COMPARISON
(1)
7656 016714 104004 401 ERROR 4 ;READ NEXT ERROR
7657 016716 000207 RTS PC ;RETURN TO "COMPAR"
7658 016720 104005 501 ERROR 5 ;WORD NOS 1 TO 4 ARE
7659 ;HEADER WORDS
7660 ;5 TO 260 ARE DATA WORDS
7661 016722 000207 RTS PC ;RETURN TO "COMPAR"
7662
7673
7674 ;*****
(3) ;*TEST 25 READ ECC ENABLED 2C
(4)
(4) ;* THIS IS AN ECC READ DATA TEST
(4) ;* ERROR CORRECTION IS ENABLED
(4) ;* A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32 AND 21
(4) ;* GOOD DATA USED IS 256 WORDS OF 17777
(4) ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD

```

```

(4) ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
(4)
(3) ;*****
(2) 016724 000004 TST25: SCOPE
7675 016726 012706 001000 MOV #STACK,SP ;RESET STACK
7676
7677
(1) 016732 012737 000025 004174 MOV #TTNO,#TSTNM ;THIS SAVES TEST NUMBER
(1)
7678
7679
7680 ; SETUP FOR WHAT IS TO BE READ
7681 ; HEADER CRC IS RESTORED FROM A SUBROUTINE
7682
7683 016740 012746 177777 MOV #1, -(SP) ;DATA TO BE READ
7684 016744 012705 000400 MOV #256,, R5 ;COUNTER
7685 016750 012700 032734 MOV #DISK, R0 ;START OF SIMULATED DISK DATA
7686 016754 011620 10: MOV (SP), (R0)+ ;MOVE IN DATA ON TO SIMULATED DISK
7687 016756 005305 DEC R5 ;COUNT
7688 016760 001375 BNE 10 ;BRANCH IF 256 NOT COMPLETE
7689 016762 005726 TST (SP)+ ;UNDO -(SP)
7690 016764 022020 CMP (R0)+,(R0)+ ;JUMP OVER THE TWO ECC WORDS
7691 016766 012705 000017 MOV #15,, R5 ;1 DATA GAP
7692 ;14 TOLERANCE GAP
7693 016772 005020 20: CLR (R0)+ ;CLEAR DATA GAP, AND
7694 016774 005305 DEC R5 ;TOLERANCE GAP
7695 016776 001375 BNE 20 ;BRANCH IF NOT COMPLETE
7696
7697
7698 017000 004737 027722 JSR PC,#FILLEC ;INSERT THE TWO ECC WORDS ON THE DISK
7699 ;IN THE CORRECT PLACE
7700
7701 ;THESE ARE FOR ECC TEST ONLY
7702
7703 017004 012737 177777 002012 MOV #1,#TSECC ;THIS IS AN ECC TEST
7704 017012 005037 027140 CLR #POSITI ;CLEAR ERROR POSITION COUNTER
7705 017016 013737 027134 027136 MOV #NCODE,#NCOUNT ;TEMPORARY N-CODE COUNTER
7706 017024 013737 027142 027150 MOV #HARDER,#HADTMP ;TEMPORARY HARD ERROR COUNTER
7707 017032 005037 027126 CLR #GECC1 ;ECC LOW ORDER TO BE GENERATED
7708 017036 005037 027130 CLR #GECC2 ;ECC HIGH ORDER TO BE GENERATED
7709 017042 005037 027144 CLR #DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
7710 017046 005037 027146 CLR #ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
7711
7712
7713 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
7714
7715 017052 012737 010000 031016 MOV #FMT22,#CYL ;16 BITS PER WORD
7716 ;CYLINDER 0, FORMAT 16 BITS
7717 017060 112737 000000 031021 MOVB #0, #SECOTR+1 ;TRACK 0
7718 017066 112737 000000 031020 MOVB #0, #SECOTR ;SECTOR 0
7719 017074 012737 000000 031022 MOV #0, #KEY1 ;KEY1=0
7720 017102 012737 000000 031024 MOV #0, #KEY2 ;KEY2=0
7721 017110 012737 000400 031076 MOV #256,, #DANORD ;NO. OF DATA WORDS
7722 017116 005037 031026 CLR #0X ;THIS IS A READ COMMAND

```



```

7723 017122 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
7724 017126 031016 CYL
7725 017130 032716 WCRC
7726
7727
7728 ;THIS IS TO INSERT ERROR
7729 ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
7730 ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
7731 ;THIS MOVE
7732 017132 012737 077757 032736 MOV 077757,00DISK+2 ;FORCE ERROR ON BIT NUMBER 32 AND 21
7733 ;SO ERROR POSITION REGISTER WILL SHOW
7734 ;22
7735 017140 012737 010040 017306 MOV 04120,,0000 ;INSERT POSITION REG.
7736
7737
7738 ;THESE ARE REGULAR SETUPS
7739 017146 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
7740 017152 012777 177374 162444 MOV 0-256,-4,,0RHWC ;256. DATA & HEADER WORDS
7741 017160 012777 003130 162440 MOV 0REINT0,0RHBA ;STARTING ADDRESS OF READ BUFFER
7742 017166 112746 000000 MOV0 00, -(SP) ;IN LOWER BYTE GET SECTOR
7743 017172 112766 000000 000001 MOV0 00, 1(SP) ;GET TRACK IN HIGHER BYTE
7744 017200 012677 162432 MOV (SP)+, 0RHDS1 ;TRACK/SECTOR IN RHDST
7745 017204 012777 010000 162430 MOV 0FMT22,0RHOF ;16 BITS PER WORD
7746 ;ECC CORRECTION NOT INHIBIT
7747 ;BECAUSE ECC IS NOT GOING
7748 ;TO BE CHECKED
7749 017212 005077 162426 CLR 0RNCA ;CYLINDER 0
7750
7751 017216 004737 024524 JSR PC, 00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
7752
7753 017222 013711 002046 MOV 00REFOR,0R1 ;READ HEADER AND DATA=72
7754 017226 005037 001774 CLR 00ERFLG0 ;CLEAR ERROR FLAG
7755 017232 004737 030706 JSR PC, 00CONHD ;READ HEADER AND DATA
7756 ;IF THERE ARE READ ERRORS THEN
7757 ;ECC WILL NOT BE CHECKED
7758
7759
7760 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7761 ;FROM THE "CONHD" ROUTINE THAT MEANS SECTOR GAP,
7762 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
7763 ;SYNC BYTE HAVE GONE BY AND SYNC'S WERE CORRECTLY
7764 ;DETECTED
7765 ;HEADER AND DATA ARE TO BE CHECKED.
7766 ;IN CHECKING READ DATA THE WRITE FROM BUFFER
7767 ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
7768 ;COMPARISONS ARE MADE
7769
7770 017236 005737 001774 TST 00ERFLG0 ;ANY ERRORS ALREADY THERE
7771
7772 (1) 017242 001103 BNE TST26 ;BRANCH IF YES
7773 (1)
7774 017244 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
7775 017250 022737 100000 001704 CMP 0DCK,00ER1 ;ONLY DATA CHECK ERROR SHOULD BE SET
7776 017256 001401 BEQ 60 ;BRANCH IF YES

```



```

7829 017426 002064          WRFROM          ;GOOD BUFFER
7830 017430 003130          REINTO         ;TEST BUFFER
7831 017432 000404          4+256,        ;NUMBER OF WORDS CHECKED
7832 017434 017442          40            ;RETURN POINT FOR ERROR HEADER
7833 017436 017446          50            ;RETURN POINT FOR ERROR DATA
7834
(1) 017440 017452          TST26         ;RETURN FOR GOOD COMPARISON
(1)
7835 017442 104004          401          ERROR 4       ;READ NEXT ERROR
7836 017444 000207          RTS          PC ;RETURN TO "COMPAR"
7837 017446 104005          501          ERROR 5       ;WORD NOS 1 TO 4 ARE
7838                                     ;HEADER WORDS
7839                                     ;5 TO 260 ARE DATA WORDS
7840 017450 000207          RTS          PC ;RETURN TO "COMPAR"
7841
7842
7843
7844
7845
7846
7855
7856          ;*****
(3)          ;TEST 26      WRITE ECC TEST ;
(4)
(4)          ;*      THIS IS A WRITE ECC TEST
(4)          ;*      WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
(4)          ;*      TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
(4)          ;*      OF ALL 52528.
(4)
(3)          ;*****
(2) 017452 000004          TST26: SCOPE
7857 017454 012706 001000          MOV          @STACK,SP ;RESET STACK
7858
7859          MOV          @TTNO,@TSTNM ;THIS SAVES TEST NUMBER
(1) 017460 012737 000026 004174
(1)
7860 017466 012700 032636          MOV          @SECGAP,R0 ;POINTER
7861 017472 012701 000460          MOV          @304,,R1 ;COUNTER
7862 017476 005020          101          CLR          (R0)+ ;CLEAR SIMULATED DISK AREA
7863 017500 005301          DEC          R1
7864 017502 001375          BNE          10 ;
7865 017504 004767 004760          JBR          PC,CLDISK ;THIS IS USED TO SET GENERAL REGISTERS
7866
7867          ;THESE ARE FOR ECC TEST ONLY
7868
7869 017510 012737 177777 002012          MOV          @-1,@TSECC ;THIS IS AN ECC TEST
7870 017516 005037 027140          CLR          @POSITI ;CLEAR ERROR POSITION COUNTER
7871 017522 013737 027134 027136          MOV          @NCODE,@NCOUNT ;TEMPORARY N-CODE COUNTER
7872 017530 013737 027142 027150          MOV          @HARDER,@HADTMP ;TEMPORARY HARD ERROR COUNTER
7873 017536 005037 027126          CLR          @GECCI ;ECC LOW ORDER TO BE GENERATED
7874 017542 005037 027130          CLR          @GECC2 ;ECC HIGH ORDER TO BE GENERATED
7875 017546 005037 027144          CLR          @DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
7876 017552 005037 027146          CLR          @ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
7877

```

```

7878
7879
7880
7881                                     ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
7882
7883 017556 012737 010000 034122      NOV      0FMT22,00WCYL      ;FORMAT22=16BIT WORDS AND
7884                                     ;CYLINDER 0
7885 017564 012737 000001 034124      NOV      01,00WSECTR      ;TRACK=0, SECTOR=1
7886 017572 005037 034126             CLR      00WKEY1          ;KEY1=0
7887 017576 005037 034130             CLR      00WKEY2          ;KEY2=0
7888 017602 012737 000400 034162      NOV      0256,,00FNWORD  ;256 DATA WORDS
7889 017610 004537 025646             JSR      R5,00CRC         ;GO TO CALCULATE CRC
7890 017614 034122
7891 017616 034132
7892
7893                                     ;THESE ARE REGULAR SETUPS
7894
7895 017620 012777 177374 161776      NOV      0-260,,0RHWC     ;256 DATA WORDS 4 HEADER WORDS
7896 017626 012700 002064             NOV      0WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
7897 017632 010077 161770             NOV      R0,0RHBA        ;ADDR. OF WRFROM INTO R0 AND
7898                                     ;BUS ADDRESS REGISTER
7899 017636 012720 010000             NOV      0FMT22,(R0)+    ;FORMAT=16 BIT WORDS
7900                                     ;CYLINDER=0
7901 017642 012720 000001             20:     NOV      01,(R0)+  ;TRACK=0, SECTOR=1, KEYS=0
7902 017646 005020                     CLR      (R0)+           ;KEY1=0
7903 017650 005020                     CLR      (R0)+           ;KEY2=0
7904 017652 012700 000400             NOV      0256,,R5        ;COUNTER
7905 017656 012720 052525             30:     NOV      052525,(R0)+ ;MOVE ALL 52525 FOR DATA
7906 017662 005305                     DEC      R5
7907 017664 001374                     BNE     30
7908 017666 012777 000001 161742      NOV      01,0RHDBT      ;BRANCH IF DATA NOT COMPLETE
7909                                     ;TRACK=0 SECTOR=1
7910
7911 (1)
7912 (1) 017674 004767 004624             JSR      PC,CHECKT      ;CHECK DVA, RDY, DPR, DRY
7913 (1)
7914 017700 013711 002042             NOV      00WRIFOR,0R1   ;GET READY FOR WRITE HEADER AND
7915                                     ;DATA WITH 62 IN RHC61
7916 017704 005037 001774             CLR      00ERFLG6       ;CLEAR ERROR FLAG
7917 017710 012777 010000 161724      NOV      0FMT22,0RHOF   ;FORMAT BIT=1 (16 BIT WORDS)
7918 017716 005077 161722             CLR      0RHCA          ;CYLINDER =0
7919 017722 004737 033776             JSR      PC,00COMWHD    ;WRITE HEADER AND DATA
7920
7921                                     ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7922                                     ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
7923                                     ;IS GOOD IE, ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
7924                                     ;ALL 52525 AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
7925                                     ;THEY ARE ALL ZEROS
7926 017726 005737 001774             TST      00ERFLG6       ;HAS ANY ERRORS OCCURED?
7927                                     ;IF WRITE ERROR OCCURS ECC IS NOT CHECKED
7928 (1) 017732 001053             BNE     TST27          ;)BRANCH IF YES

```

(1)
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
(1)
(1)
7970
7971
7972
7973
7974
7975
7976
7977
7978

017734 023737 027126 033734
017742 001402
017744 104031
017746 000405
017750 023737 027130 033736 681
017756 001401
017760 104031

017762 004737 024706 781

017766 004037 024406
017772 003130
017774 004126
017776 052525

020000 013737 027126 004130
020006 013737 027130 004132
020014 004037 024406
020020 004134
020022 004170
020024 000000

020026 005037 001774

020032 004037 025342
020036 003130
020040 032734
020042 000402
020044 020052
020046 020056
020050 020062
020052 104007 481
020054 000207
020056 104010 581

```

;COMPARE SOFTWARE GENERATED ECC WITH THAT GENERATED BY HARDWARE
CMP      00GECC1,00WECC1;COMPARE SOFTWARE ECC WITH HARDWARE ECC
BEQ      68              ;BRANCH IF GOOD
ERROR    31              ;LOW ORDER ECC IN ERROR
BR       78              ;BRANCH TO CONTINUE
CMP      00GECC2,00WECC2;COMPARE SOFTWARE ECC WITH HARDWARE ECC
BEQ      78              ;BRANCH IF GOOD
ERROR    31              ;HIGH ORDER ECC IN ERROR

781      JSR      PC,00CHECKE ;CHECK DVA,RDY,DRY,DPR

;FILL "REINTO" BUFFER WITH EXPECTED DATA
JSR      R0,00CLAREA ;FILL REINTO BUFFER
REINTO
REINTO+<255,*2> ;FROM
;WORD 52528 ;DATA ;TO

MOV      00GECC1,00REINTO+<256,*2>;FILL ECC1
MOV      00GECC2,00REINTO+<257,*2>;FILL ECC2
JSR      R0,00CLAREA ;FILL REST
REINTO+<258,*2> ;FROM
REINTO+<272,*2> ;TO
0 ;DATA

CLR      00ERFLG8 ;CLEAR ERROR FLAG

;NOW COMPARE "DISK" BUFFER WITH "REINTO"
JSR      R0,00COMPAR ;CHECK
REINTO ;GOOD BUFFER
DISK ;TEST BUFFER
258, ;NUMBER OF WORDS CHECKED
48 ;RETURN POINT FOR ERROR HEADER
58 ;RETURN POINT FOR ERROR DATA

TST27 ;RETURN FOR GOOD COMPARISON

481      ERROR    7 ;READ ERROR IS NEXT
RTS      PC ;RETURN TO COMPARE
581      ERROR    10 ;WORD NOS 1 TO 256 ARE
;DATA WORDS
;WORD NOS 257 AND 258
;ARE ECC WHICH ARE CHECKED
;WORD NOS 259
;IS DATA GAP
;WORD NOS 260 TO 273
    
```

```
7979                                     ;ARE TOLERANCE GAP
7980 020060 000207                       RTS    PC    ;RETURN TO COMPARE
7981
7982
7983
7984
7985
7986
7987
7988                                     ;*****
7989                                     ;TEST 27    READ ECC ENABLED JA
7990                                     ;*****
7991                                     ;*
7992                                     ;* THIS IS AN ECC READ DATA TEST
7993                                     ;* ERROR CORRECTION IS ENABLED
7994                                     ;* NO ERROR IS INSERTED
7995                                     ;* GOOD DATA USED IS 256 WORDS OF 52525
7996                                     ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
7997                                     ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
7998                                     ;*****
7999 TST27: SCOPE
8000     MOV    0STACK,SP    ;RESET STACK
8001     MOV    0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
8002
8003                                     ;
8004                                     ; SETUP FOR WHAT IS TO BE READ
8005                                     ; HEADER CRC IS RESTORED FROM A SUBROUTINE
8006     MOV    052525, -(SP) ;DATA TO BE READ
8007     MOV    0256,, R5     ;COUNTER
8008     MOV    0DISK, R0     ;START OF SIMULATED DISK DATA
8009     MOV    (SP), (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
8010     DEC    R5           ;COUNT
8011     BNE    10          ;BRANCH IF 256 NOT COMPLETE
8012     TST    (SP)+       ;UNDO -(SP)
8013     CMP    (R0)+,(R0)+ ;JUMP OVER THE TWO ECC WORDS
8014     MOV    015,, R5    ;1 DATA GAP
8015                                     ;14 TOLERANCE GAP
8016     CLR    (R0)+       ;CLEAR DATA GAP, AND
8017     DEC    R5         ;TOLERANCE GAP
8018     BNE    20         ;BRANCH IF NOT COMPLETE
8019
8020
8021     JSR    PC,00FILLEC ;INSERT THE TWO ECC WORDS ON THE DISK
8022                                     ;IN THE CORRECT PLACE
8023
8024                                     ;THESE ARE FOR ECC TEST ONLY
8025
8026     MOV    0-1,00TSECC ;THIS IS AN ECC TEST
8027     CLR    00POSITI    ;CLEAR ERROR POSITION COUNTER
8028     MOV    00NCODE,00NCOUNT ;TEMPORARY N-CODE COUNTER
8029     MOV    00HARDER,00HADTMP ;TEMPORARY HARD ERROR COUNTER
```

```

0030 020170 005037 027126 CLR 00GECC1 ;ECC LOW ORDER TO BE GENERATED
0031 020174 005037 027130 CLR 00GECC2 ;ECC HIGH ORDER TO BE GENERATED
0032 020200 005037 027144 CLR 00DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
0033 020204 005037 027146 CLR 00ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
0034
0035
0036 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
0037
0038 020210 012737 010000 031016 MOV 0FMT22,00CYL ;16 BITS PER WORD
0039 ;CYLINDER 0, FORMAT 16 BITS
0040 020216 112737 000000 031021 MOVB 00, 00SECOTR+1 ;TRACK 0
0041 020224 112737 000000 031020 MOVB 00, 00SECOTR ;SECTOR 0
0042 020232 012737 000000 031022 MOV 00, 00KEY1 ;KEY1=0
0043 020240 012737 000000 031024 MOV 00, 00KEY2 ;KEY2=0
0044 020246 012737 000400 031076 MOV 0256,, 00DANORD ;NO. OF DATA WORDS
0045 020254 005037 031026 CLR 00X ;THIS IS A READ COMMAND
0046 020260 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
0047 020264 031016 CYL
0048 020266 032716 WCRC
0049
0050
0051
0052
0053 ;THESE ARE REGULAR SETUPS
0054 020270 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
0055 020274 012777 177374 161322 MOV 0-256,-4,,0RHWC ;256, DATA 4 HEADER WORDS
0056 020302 012777 003130 161316 MOV 0REINT0,0RHBA ;STARTING ADDRESS OF READ BUFFER
0057 020310 112746 000000 MOVB 00, -(SP) ;IN LOWER BYTE GET SECTOR
0058 020314 112766 000000 000001 MOVB 00, 1(SP) ;GET TRACK IN HIGHER BYTE
0059 020322 012677 161310 MOV (SP)+, 0RHDS1 ;TRACK/SECTOR IN RHDST
0060 020326 012777 010000 161306 MOV 0FMT22,0RHOF ;16 BITS PER WORD
0061 ;ECC CORRECTION NOT INHIBIT
0062 ;BECAUSE ECC IS NOT GOING
0063 ;TO BE CHECKED
0064 020334 005077 161304 CLR 0RHCA ;CYLINDER 0
0065
0066 020340 004737 024524 JSR PC, 00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
0067
0068 020344 013711 002046 MOV 00REFOR,0R1 ;READ HEADER AND DATA=72
0069 020350 005037 001774 CLR 00ERFLG8 ;CLEAR ERROR FLAG
0070 020354 004737 030706 JSR PC, 00COMHD ;READ HEADER AND DATA
0071 ;IF THERE ARE READ ERRORS THEN
0072 ;ECC WILL NOT BE CHECKED
0073
0074
0075 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
0076 ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
0077 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
0078 ;SYNC BYTE HAVE GONE BY AND SYNC8 WERE CORRECTLY
0079 ;DETECTED
0080 ;HEADER AND DATA ARE TO BE CHECKED,
0081 ;IN CHECKING READ DATA THE WRITE FROM BUFFER
0082 ;"NRFROM" IS FILLED WITH EXPECTED DATA AND
0083 ;COMPARISONS ARE MADE

```

```

0084
0085 020360 005737 001774          TST      00ERFLG8          ;ANY ERRORS ALREADY THERE
0086
(1) 020364 001077          BNE      TST30          ;BRANCH IF YES
(1)
0087 020366 004737 024170          JSR      PC,00PUTREG      ;SAVE REGISTERS
0088 020372 005737 001704          TST      00ER1          ;NO ERRORS SHOULD BE SET
0089 020376 001401          BEQ      68          ;BRANCH IF NO ERRORS SET
0090 020400 104032          ERROR   32          ;32 BIT ECC REGISTER SHOULD BE ZERO
0091
0092
0093
0094 020402 013746 027126          68:     NOV      00GECC1,-(SP) ;GET PATTERN REGISTER
0095 020406 042716 174000          BIC      0174000,(SP) ;KEEP ONLY 11 BITS
0096 020412 022637 001734          CMP      (SP)+,00EC2 ;COMPARE PATTERN REGISTER
0097 020416 001401          BEQ      78          ;BRANCH IF GOOD
0098 020420 104032          ERROR   32          ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
0099
0100
0101
0102
0103
0104
0105
0106 020422 012700 000020          78:     NOV      016,,R0          ;COUNTER
0107 020426 052777 000002 161216 88:     BIS      0MCLK,0RHMR ;SET CLOCK
0108 020434 042777 000002 161210          BIC      0MCLK,0RHMR ;CLEAR CLOCK
0109 020442 005300          DEC      R0          ;COUNT
0110 020444 001370          BNE      88          ;BRANCH IF 16 CLOCKS NOT DONE
0111 020446 004737 024706          JSR      PC,00CHECKE ;CHECK DVA,DRY,RDY,DPR
0112 020452 012700 002064          NOV      0WRFRON,R0 ;GETTING READY TO FILL EXPECTED DATA
0113 020456 012720 010000          NOV      00IFMT22,(R0)+ ;CYLINDER 0
0114 020462 112746 000000          NOV      00, -(SP) ;IN LOWER BYTE GET SECTOR
0115 020466 112766 000000 000001          NOV      00, 1(SP) ;GET TRACK IN HIGHER BYTE
0116 020474 012620          NOV      (SP)+, (R0)+ ;GET TRACK/SECTOR IN BUFFER
0117 020476 012720 000000          NOV      00, (R0)+ ;KEY1 IN BUFFER
0118 020502 012720 000000          NOV      00, (R0)+ ;KEY2 IN BUFFER
0119 020506 012701 000400          NOV      0256,, R1 ;DATA WORD COUNTER
0120 020512 012702 052525          NOV      052525, R2 ;DATA
0121 020516 010220          38:     NOV      R2, (R0)+ ;DATA INTO BUFFER
0122 020520 005301          DEC      R1          ;COUNT
0123 020522 001375          BNE      38          ;BRANCH IF 256 NOT DONE
0124
0125
0126
0127
0128 020524 005037 001774          CLR      00ERFLG8          ;CLEAR ERROR FLAG
0129
0130 020530 004737 024170          JSR      PC,00PUTREG      ;SAVE REGISTERS
0131
0132
0133
0134 020534 004037 025342          JSR      R0,00COMPAR ;CHECK
0135 020540 002064          WRFRON ;GOOD BUFFER

```



```

0136 020542 003130          REINTO          ;TEST BUFFER
0137 020544 000404          4+256,        ;NUMBER OF WORDS CHECKED
0138 020546 020554          48            ;RETURN POINT FOR ERROR HEADER
0139 020550 020560          50            ;RETURN POINT FOR ERROR DATA
0140
(1) 020552 020564          TST30         ;RETURN FOR GOOD COMPARISON
(1)
0141 020554 104004          401 ERROR 4    ;READ NEXT ERROR
0142 020556 000207          RTS PC        ;RETURN TO "COMPAR"
0143 020560 104005          501 ERROR 5    ;WORD NOS 1 TO 4 ARE
0144                                     ;HEADER WORDS
0145                                     ;5 TO 260 ARE DATA WORDS
0146 020562 000207          RTS PC        ;RETURN TO "COMPAR"
0147
0148
0149
0150
0162
0163 ;*****
(3) ;TEST 30 READ ECC ENABLED 3B
(4)
(4) ;* THIS IS AN ECC READ DATA TEST
(4) ;* ERROR CORRECTION IS ENABLED
(4) ;* A CORRECTABLE ERROR IS INSERTED IN BIT POSITION 4120
(4) ;* THIS IS THE LAST BIT OF THE ECC
(4) ;* GOOD DATA USED IS 256 WORDS OF 52525
(4) ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
(4) ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
(4)
(3) ;*****
(2) 020564 000004          TST30: SCOPE
0164 020566 012706 001000          MOV 0STACK,SP ;RESET STACK
0165
0166
(1) 020572 012737 000030 004174          MOV 0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
0167
0168
0169 ; SETUP FOR WHAT IS TO BE READ
0170 ; HEADER CRC IS RESTORED FROM A SUBROUTINE
0171
0172 020600 012746 052525          MOV 052525, -(SP) ;DATA TO BE READ
0173 020604 012705 000400          MOV 0256,, R5    ;COUNTER
0174 020610 012700 032734          MOV 0DISK, R0    ;START OF SIMULATED DISK DATA
0175 020614 011620          101 MOV (SP), (R0)+ ;MOVE IN DATA ON TO SIMULATED DISK
0176 020616 005305          DEC R5           ;COUNT
0177 020620 001375          BNE 10          ;BRANCH IF 256 NOT COMPLETE
0178 020622 005726          TST (SP)+       ;UNDO -(SP)
0179 020624 022020          CMP (R0)+,(R0)+ ;JUMP OVER THE TWO ECC WORDS
0180 020626 012705 000017          MOV 015,, R5    ;1 DATA GAP
0181                                     ;14 TOLERANCE GAP
0182 020632 005020          201 CLR (R0)+      ;CLEAR DATA GAP, AND
0183 020634 005305          DEC R5           ;TOLERANCE GAP
0184 020636 001375          BNE 20          ;BRANCH IF NOT COMPLETE

```

```

0185
0186
0187 020640 004737 027722 JSR PC,0*FILLEC ;INSERT ECC IN PROPER PLACE ON DISK
0188
0189
0190
0191 ;THESE ARE FOR ECC TEST ONLY
0192
0193 020644 012737 177777 002012 MOV 0-1,0*TSECC ;THIS IS AN ECC TEST
0194 020652 005037 027140 CLR 0*POSITI ;CLEAR ERROR POSITION COUNTER
0195 020656 013737 027134 027136 MOV 0*NCODE,0*NCOUNT ;TEMPORARY N-CODE COUNTER
0196 020664 013737 027142 027150 MOV 0*HARDER,0*HADTMP ;TEMPORARY HARD ERROR COUNTER
0197 020672 005037 027126 CLR 0*GECC1 ;ECC LOW ORDER TO BE GENERATED
0198 020676 005037 027130 CLR 0*GECC2 ;ECC HIGH ORDER TO BE GENERATED
0199 020702 005037 027144 CLR 0*DATENV ;CLEAR DATA ENVELOPE CLOCK COUNT
0200 020706 005037 027146 CLR 0*ZCODE ;CLEAR LEADING ZEROS CLOCK COUNT
0201
0202
0203 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
0204
0205 020712 012737 010000 031016 MOV 0*FMT22,0*CYL ;16 BITS PER WORD
0206 ;CYLINDER 0, FORMAT 16 BITS
0207 020720 112737 000000 031021 MOV 0, 0*SECOTR+1 ;TRACK 0
0208 020726 112737 000000 031020 MOV 0, 0*SECOTR ;SECTOR 0
0209 020734 012737 000000 031022 MOV 0, 0*KEY1 ;KEY1=0
0210 020742 012737 000000 031024 MOV 0, 0*KEY2 ;KEY2=0
0211 020750 012737 000400 031076 MOV 0256,, 0*DAWORD ;NO. OF DATA WORDS
0212 020756 005037 031026 CLR 0*X ;THIS IS A READ COMMAND
0213 020762 004537 025646 JSR R5,0*CRC ;GO TO CALCULATE CRC
0214 020766 031016 CYL
0215 020770 032716 WCRC
0216
0217
0218 ;THIS IS TO INSERT ERROR
0219 ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
0220 ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
0221 ;THIS MOVE
0222 ;THIS CHANGES THE LAST BIT OF THE ECC
0223 020772 013746 033736 MOV 0*WECC2,-(SP) ;GET LAST ECC
0224 020776 005116 COM (SP) ;INVERT ALL BITS OF WECC2
0225 021000 042716 077777 BIC 0*C100000,(SP) ;KEEP BIT 16
0226 021004 042737 100000 033736 BIC 0100000,0*WECC2 ;CLEAR BIT 16 IN ECC
0227 021012 052637 033736 BIS (SP)+,0*WECC2 ;THIS WILL SET BIT 16 IF IT WAS 0
0228 ;OR WILL SET NOTHING IF IT WAS A 1
0229
0230
0231 021016 012737 010026 021164 MOV 04110,,0000 ;INSERT POSITION REG.
0232
0233
0234 ;THESE ARE REGULAR SETUPS
0235 021024 004737 024470 JSR PC,0*CLDISK ;SETUP GENERAL REGISTERS
0236 021030 012777 177374 160566 MOV 0-256,-4,,0*RHWC ;256, DATA 4 HEADER WORDS
0237 021036 012777 003130 160562 MOV 0*REINTO,0*RHBA ;STARTING ADDRESS OF READ BUFFER
0238 021044 112746 000000 MOV 0, -(SP) ;IN LOWER BYTE GET SECTOR

```

```

0239 021050 112766 000000 000001      MOVB 00, 1(SP) ;GET TRACK IN HIGHER BYTE
0240 021056 012677 160554             MOV (SP)+, 0RHDST ;TRACK/SECTOR IN RHDST
0241 021062 012777 010000 160552      MOV 0FMT22,0RHOF ;16 BITS PER WORD
0242                                     ;ECC CORRECTION NOT INHIBIT
0243                                     ;BECAUSE ECC IS NOT GOING
0244                                     ;TO BE CHECKED
0245 021070 005077 160550             CLR 0RHCA ;CYLINDER 0
0246
0247 021074 004737 024524             JSR PC, 00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
0248
0249 021100 013711 002046             MOV 00REFOR,0R1 ;READ HEADER AND DATA=72
0250 021104 005037 001774             CLR 00ERFLG0 ;CLEAR ERROR FLAG
0251 021110 004737 030706             JSR PC, 00COMHD ;READ HEADER AND DATA
0252                                     ;IF THERE ARE READ ERRORS THEN
0253                                     ;ECC WILL NOT BE CHECKED
0254
0255
0256                                     ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
0257                                     ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
0258                                     ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
0259                                     ;SYNC BYTE HAVE GONE BY AND SYNCB WERE CORRECTLY
0260                                     ;DETECTED
0261                                     ;HEADER AND DATA ARE TO BE CHECKED,
0262                                     ;IN CHECKING READ DATA THE WRITE FROM BUFFER
0263                                     ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
0264                                     ;COMPARISONS ARE MADE
0265
0266 021114 005737 001774             TST 00ERFLG0 ;ANY ERRORS ALREADY THERE
0267
0268 (1) 021120 001071                 BNE TST31 ;BRANCH IF YES
0269 (1)
0268 021122 004737 024170             JSR PC,00PUTREG ;SAVE REGISTERS
0269 021126 022737 100000 001704      CMP 0DCK,00ER1 ;ONLY DATA CHECK ERROR SHOULD BE SET
0270 021134 001401                 BEQ 60 ;BRANCH IF YES
0271 021136 104032                 ERROR 32 ;32 BIT ECC REGISTER SHOULD BE NON
0272                                     ;ZERO
0273                                     ;ONLY 11 OF THE 32 BITS CAN BE SEEN
0274                                     ;IN THE PATERN REGISTER
0275                                     ;DCK SHOULD BE SET IN RHER1
0276 021140 013746 027126             MOV 00GECC1,-(SP) ;GET PATTERN REGISTER
0277 021144 042716 174000             BIC 0174000,(SP) ;KEEP ONLY 11 BITS
0278 021150 022637 001734             CMP (SP)+,00EC2 ;COMPARE PATTERN REGISTER
0279 021154 001401                 BEQ 70 ;BRANCH IF GOOD
0280 021156 104032                 ERROR 32 ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
0281
0282 021160 004037 027550             JSR R0,00ECORR ;GO TO ECC CORRECTION PROCESS
0283 021164 010026                 001 4110, ;EXPECTED POSITION REG, WHEN CORRECTION
0284                                     ;IS COMPLETE
0285
0286
0287
0288 021166 004737 024706             JSR PC,00CHECKE ;CHECK DVA,DRY,RDY,DPR
0289 021172 012700 002064             MOV 0WRFROM,R0 ;GETTING READY TO FILL EXPECTED DATA
0290 021176 012720 010000             MOV 00IFMT22,(R0)+ ;CYLINDER 0

```

```
0291 021202 112746 000000      MOVB  00,    -(SP)    ;IN LOWER BYTE GET SECTOR
0292 021206 112766 000000 000001  MOVB  00,    1(SP)    ;GET TRACK IN HIGHER BYTE
0293 021214 012620      MOV    (SP)+, (R0)+   ;GET TRACK/SECTOR IN BUFFER
0294 021216 012720 000000      MOV    00,    (R0)+   ;KEY1 IN BUFFER
0295 021222 012720 000000      MOV    00,    (R0)+   ;KEY2 IN BUFFER
0296 021226 012701 000400      MOV    0256,, R1     ;DATA WORD COUNTER
0297 021232 012702 052525      MOV    052525, R2    ;DATA
0298 021236 010220      36:  MOV    R2,    (R0)+   ;DATA INTO BUFFER
0299 021240 005301      DEC    R1            ;COUNT
0300 021242 001375      BNE    30            ;BRANCH IF 256 NOT DONE
0301
0302                                ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
0303                                ;NOW THE INSERTED ERROR WILL BE PUT IN
0304                                ;BUT INSERTED ERROR IS IN ECC SO DATA IS NOT WRONG
0305
0306
0307 021244 004737 024170      JSR    PC,00PUTREG   ;SAVE REGISTERS
0308
0309 021250 005037 001774      CLR    00ERFLG0     ;CLEAR ERROR FLAG
0310
0311
0312                                ;NOW READ DATA BUFFER WILL BE CHECKED
0313
0314 021254 004037 025342      JSR    R0,00COMPAR   ;CHECK
0315 021260 002064      WRFROM              ;GOOD BUFFER
0316 021262 003130      REINTO              ;TEST BUFFER
0317 021264 000404      4+256,              ;NUMBER OF WORDS CHECKED
0318 021266 021274      40                  ;RETURN POINT FOR ERROR HEADER
0319 021270 021300      50                  ;RETURN POINT FOR ERROR DATA
0320
(1) 021272 021304      TST31              ;RETURN FOR GOOD COMPARISON
(1)
0321 021274 104004      40:  ERROR  4        ;READ NEXT ERROR
0322 021276 000207      RTS    PC           ;RETURN TO "COMPAR"
0323 021300 104005      50:  ERROR  5        ;WORD NOS 1 TO 4 ARE
0324                                ;HEADER WORDS
0325                                ;5 TO 260 ARE DATA WORDS
0326 021302 000207      RTS    PC           ;RETURN TO "COMPAR"
0327
0339
0340                                ;*****
(3)                                ;*TEST 31 READ ECC ENABLED 3C
(4)
(4)                                ;* THIS IS AN ECC READ DATA TEST
(4)                                ;* ERROR CORRECTION IS ENABLED
(4)                                ;* A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 296 THRU 300
(4)                                ;* THIS IS IN WORD NUMBER 19 AND 20
(4)                                ;* GOOD DATA USED IS 256 WORDS OF 52525
(4)                                ;* COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
(4)                                ;* TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
(4)
(3)                                ;*****
(2) 021304 000004      TST31: SCOPE
0341 021306 012706 001000      NOV    0STACK,SP    ;RESET STACK
```

```

0342
0343 (1) 021312 012737 000031 004174      MOV      @TTNO,@TSTNM      ;THIS SAVES TEST NUMBER
0344 (1)
0345
0346 ;          SETUP FOR WHAT IS TO BE READ
0347 ;          HEADER CRC IS RESTORED FROM A SUBROUTINE
0348
0349 021320 012746 052525      MOV      @52525, -(SP)    ;DATA TO BE READ
0350 021324 012705 000400      MOV      @256,, R5       ;COUNTER
0351 021330 012700 032734      MOV      @DISK, R0       ;START OF SIMULATED DISK DATA
0352 021334 011620          10:  MOV      (SP), (R0)+     ;MOVE IN DATA ON TO SIMULATED DISK
0353 021336 005305          DEC      R5              ;COUNT
0354 021340 001375          BNE     10              ;BRANCH IF 256 NOT COMPLETE
0355 021342 005726          TST     (SP)+          ;UNDO -(SP)
0356 021344 022020          CMP     (R0)+,(R0)+    ;JUMP OVER THE TWO ECC WORDS
0357 021346 012705 000017      MOV      @15,, R5        ;1 DATA GAP
0358                                ;14 TOLERANCE GAP
0359 021352 005020          20:  CLR     (R0)+          ;CLEAR DATA GAP, AND
0360 021354 005305          DEC     R5              ;TOLERANCE GAP
0361 021356 001375          BNE     20              ;BRANCH IF NOT COMPLETE
0362
0363
0364 021360 004737 027722      JSR     PC,@FILLEC      ;INSERT THE TWO ECC WORDS ON THE DISK
0365                                ;IN THE CORRECT PLACE
0366
0367 ;THESE ARE FOR ECC TEST ONLY
0368
0369 021364 012737 177777 002012      MOV      @-1,@TSECC      ;THIS IS AN ECC TEST
0370 021372 005037 027140      CLR     @POSITI         ;CLEAR ERROR POSITION COUNTER
0371 021376 013737 027134 027136      MOV      @NCODE,@NCOUNT ;TEMPORARY N-CODE COUNTER
0372 021404 013737 027142 027150      MOV      @HARDER,@HADTMP  ;TEMPORARY HARD ERROR COUNTER
0373 021412 005037 027126      CLR     @GECC1          ;ECC LOW ORDER TO BE GENERATED
0374 021416 005037 027130      CLR     @GECC2          ;ECC HIGH ORDER TO BE GENERATED
0375 021422 005037 027144      CLR     @DATENV         ;CLEAR DATA ENVELOPE CLOCK COUNT
0376 021426 005037 027146      CLR     @ZCODE          ;CLEAR LEADING ZEROS CLOCK COUNT
0377
0378
0379 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
0380
0381 021432 012737 010000 031016      MOV      @FMT22,@CYL     ;16 BITS PER WORD
0382                                ;CYLINDER 0, FORMAT 16 BITS
0383 021440 112737 000000 031021      MOV     @0, @SECOTR+1 ;TRACK 0
0384 021446 112737 000000 031020      MOV     @0, @SECOTR ;SECTOR 0
0385 021454 012737 000000 031022      MOV     @0, @KEY1 ;KEY1=0
0386 021462 012737 000000 031024      MOV     @0, @KEY2 ;KEY2=0
0387 021470 012737 000400 031076      MOV     @256,, @DAWORD ;NO. OF DATA WORDS
0388 021476 005037 031026      CLR     @X              ;THIS IS A READ COMMAND
0389 021502 004537 025646      JSR     R5,@CRC         ;GO TO CALCULATE CRC
0390 021506 031016      CYL
0391 021510 032716      WCRC
0392
0393

```

```

0394 ;THIS IS TO INSERT ERROR
0395 ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
0396 ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
0397 ;THIS MOVE
0398 021512 012737 152652 033000 MOV 0152652,00DISK+44,INSERT ERROR IN POSITION 296 THRU 304
0399 ;IN WORD NUMBER 19
0400 021520 012737 052532 033002 MOV 052532,00DISK+46,INSERT ERROR IN POSITION 305 THRU 308
0401 ;IN WORD NUMBER 20
0402 021526 012737 010040 021674 MOV 04120,,0000 ;INSERT POSITION REG.
0403
0404
0405 ;THESE ARE REGULAR SETUPS
0406 021534 004737 024470 JSR PC,00CLDISK ;SETUP GENERAL REGISTERS
0407 021540 012777 177374 160056 MOV 0-256,-4,,0RHWC ;256, DATA 4 HEADER WORDS
0408 021546 012777 003130 160052 MOV 0REINT0,0RHBA ;STARTING ADDRESS OF READ BUFFER
0409 021554 112746 000000 MOV 00, -(SP) ;IN LOWER BYTE GET SECTOR
0410 021560 112766 000000 000001 MOV 00, 1(SP) ;GET TRACK IN HIGHER BYTE
0411 021566 012677 160044 MOV (SP)+, 0RHDBT ;TRACK/SECTOR IN RHDST
0412 021572 012777 010000 160042 MOV 0FMT22,0RHOP ;16 BITS PER WORD
0413 ;ECC CORRECTION NOT INHIBIT
0414 ;BECAUSE ECC IS NOT GOING
0415 ;TO BE CHECKED
0416 021600 005077 160040 CLR 0RHCA ;CYLINDER 0
0417
0418 021604 004737 024524 JSR PC, 00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
0419
0420 021610 013711 002046 MOV 00REFOR,0R1 ;READ HEADER AND DATA=72
0421 021614 005037 001774 CLR 00ERFLG0 ;CLEAR ERROR FLAG
0422 021620 004737 030706 JSR PC, 00COMHD ;READ HEADER AND DATA
0423 ;IF THERE ARE READ ERRORS THEN
0424 ;ECC WILL NOT BE CHECKED
0425
0426
0427 ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
0428 ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
0429 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
0430 ;SYNC BYTE HAVE GONE BY AND SYNCB WERE CORRECTLY
0431 ;DETECTED
0432 ;HEADER AND DATA ARE TO BE CHECKED,
0433 ;IN CHECKING READ DATA THE WRITE FROM BUFFER
0434 ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
0435 ;COMPARISONS ARE MADE
0436
0437 021624 005737 001774 TST 00ERFLG0 ;ANY ERRORS ALREADY THERE
0438
0439 (1) 021630 001106 BNE TST32 ;BRANCH IF YES
0440 (1)
0439 021632 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
0440 021636 022737 100000 001704 CMP 0DCK,00ER1 ;ONLY DATA CHECK ERROR SHOULD BE SET
0441 021644 001401 BEQ 66 ;BRANCH IF YES
0442 021646 104032 ERROR 32 ;32 BIT ECC REGISTER SHOULD BE NON
0443 ;ZERO
0444 ;ONLY 11 OF THE 32 BITS CAN BE SEEN
0445 ;IN THE PATERN REGISTER

```

```

0446
0447 021650 013746 027126      681  MOV      00GECC1,-(SP)      ;DCK SHOULD BE SET IN RHER1
0448 021654 042716 174000      BIC      0174000,(SP)      ;GET PATTERN REGISTER
0449 021660 022637 001734      CMP      (SP)+,00EC2      ;KEEP ONLY 11 BITS
0450 021664 001401      BEQ      78                ;COMPARE PATTERN REGISTER
0451 021666 104032      ERROR   32                ;BRANCH IF GOOD
0452                                     ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
0453 021670 004037 027550      781  JSR      R0,00ECORR      ;GO TO ECC CORRECTION PROCESS
0454 021674 000000      881  ,WORD                    ;EXPECTED POSITION REG, WHEN CORRECTION
0455                                     ;IS COMPLETE
0456
0457
0458
0459 021676 004737 024170      JSR      PC,00PUTREG      ;SAVE REGISTERS
0460 021702 022737 100100 001704      CMP      0DCKIECH,00ER1   ;WITH ERRORS INSERTED IN BIT POSITION 21
0461                                     ;THRU 32 HARD ERROR BIT SHOULD SET
0462 021710 001401      BEQ      98                ;BRANCH IF GOOD
0463 021712 104036      ERROR   36                ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
0464                                     ;32 HCE SHOULD SET
0465
0466
0467
0468 021714 004737 024706      981  JSR      PC,00CHECKE     ;CHECK DVA,DRY,RDY,DPR
0469 021720 012700 002064      MOV      0WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
0470 021724 012720 010000      MOV      00IFMT22,(R0)+   ;CYLINDER 0
0471 021730 112746 000000      MOVVB   00,-(SP)         ;IN LOWER BYTE GET SECTOR
0472 021734 112766 000000 000001      MOVVB   00,1(SP)         ;GET TRACK IN HIGHER BYTE
0473 021742 012620      MOV      (SP)+,(R0)+      ;GET TRACK/SECTOR IN BUFFER
0474 021744 012720 000000      MOV      00,(R0)+        ;KEY1 IN BUFFER
0475 021750 012720 000000      MOV      00,(R0)+        ;KEY2 IN BUFFER
0476 021754 012701 000400      MOV      0256,,R1         ;DATA WORD COUNTER
0477 021760 012702 052525      MOV      052525,R2        ;DATA
0478 021764 010220      301  MOV      R2,(R0)+        ;DATA INTO BUFFER
0479 021766 005301      DEC      R1                ;COUNT
0480 021770 001375      BNE     30                ;BRANCH IF 256 NOT DONE
0481
0482                                     ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
0483                                     ;NOW THE INSERTED ERROR WILL BE PUT IN
0484 021772 012737 152652 002140      MOV      0152652,00WRFROM+54;INSERT ERROR IN POSITION 296 THRU 304
0485                                     ;IN WORD NUMBER 19 IN DATA
0486 022000 012737 052532 002142      MOV      052532,00WRFROM+56;INSERT ERROR IN POSITION 305 THRU 308
0487                                     ;IN WORD NUMBER 20 IN DATA
0488
0489 022006 004737 024170      JSR      PC,00PUTREG      ;SAVE REGISTERS
0490
0491
0492 022012 005037 001774      CLR      00ERFLG0         ;CLEAR ERROR FLAG
0493
0494
0495                                     ;NOW READ DATA BUFFER WILL BE CHECKED
0496
0497 022016 004037 025342      JSR      R0,00COMPAR      ;CHECK
0498 022022 002064      WRFROM                    ;GOOD BUFFER
0499 022024 003130      REINTO                    ;TEST BUFFER

```

```

0500 022026 000404          4+256,          ;NUMBER OF WORDS CHECKED
0501 022030 022036          48              ;RETURN POINT FOR ERROR HEADER
0502 022032 022042          58              ;RETURN POINT FOR ERROR DATA
0503
(1) 022034 022046          TST32           ;RETURN FOR GOOD COMPARISON
(1)
0504 022036 104004          48:  ERROR 4     ;READ NEXT ERROR
0505 022040 000207          RTS  PC         ;RETURN TO "COMPAR"
0506 022042 104005          58:  ERROR 5     ;WORD NOS 1 TO 4 ARE
0507                                     ;HEADER WORDS
0508                                     ;5 TO 260 ARE DATA WORDS
0509 022044 000207          RTS  PC         ;RETURN TO "COMPAR"
0510
0522
0523 ;*****
(3) ;TEST 32      READ ECC ENABLED 3D
(4)
(4) ;*      THIS IS AN ECC READ DATA TEST
(4) ;*      ERROR CORRECTION IS ENABLED
(4) ;*      A NON CORRECTABLE ERROR IS INSERTED IN BIT POSITION 32 AND 4096
(4) ;*      4096 IS THE LAST DATA BIT
(4) ;*      GOOD DATA USED IS 256 WORDS OF 52525
(4) ;*      COMMAND IS GIVEN FOR CYLINDER 0 FORMAT 16 BITS PER WORD
(4) ;*      TRACK 0, SECTOR 0 KEYS 0 READ HEADER AND DATA
(3) ;*****
(2) 022046 000004          TST32:  SCOPE
0524 022050 012706 001000  MOV      0STACK,SP      ;RESET STACK
0525
0526 (1) 022054 012737 000032 004174  MOV      0TTNO,0TSTNM   ;THIS SAVES TEST NUMBER
(1)
0527
0528
0529 ;      SETUP FOR WHAT IS TO BE READ
0530 ;      HEADER CRC IS RESTORED FROM A SUBROUTINE
0531
0532 022062 012746 052525  MOV      052525, -(SP)   ;DATA TO BE READ
0533 022066 012705 000400  MOV      0256,, R5      ;COUNTER
0534 022072 012700 032734  MOV      0DISK, R0      ;START OF SIMULATED DISK DATA
0535 022076 011620          18:  MOV      (SP), (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
0536 022100 005305          DEC      R5              ;COUNT
0537 022102 001375          BNE     18              ;BRANCH IF 256 NOT COMPLETE
0538 022104 005726          TST     (SP)+          ;UNDO -(SP)
0539 022106 022020          CMP     (R0)+,(R0)+   ;JUMP OVER THE TWO ECC WORDS
0540 022110 012705 000017  MOV      015,, R5      ;1 DATA GAP
0541
0542 022114 005020          28:  CLR     (R0)+       ;14 TOLERANCE GAP
0543 022116 005305          DEC     R5              ;CLEAR DATA GAP, AND
0544 022120 001375          BNE     28              ;TOLERANCE GAP
0545                                     ;BRANCH IF NOT COMPLETE
0546
0547 022122 004737 027722  JBR     PC,00FILLEC    ;INSERT THE TWO ECC WORDS ON THE DISK
0548                                     ;IN THE CORRECT PLACE

```



```

0549
0550                                     ;THESE ARE FOR ECC TEST ONLY
0551
0552 022126 012737 177777 002012      MOV     0-1,00TSECC                ;THIS IS AN ECC TEST
0553 022134 005037 027140              CLR     00POSITI                   ;CLEAR ERROR POSITION COUNTER
0554 022140 013737 027134 027136      MOV     00NCODE,00NCOUNT           ;TEMPORARY N-CODE COUNTER
0555 022146 013737 027142 027150      MOV     00HARDER,00HADTMP         ;TEMPORARY HARD ERROR COUNTER
0556 022154 005037 027126              CLR     00GECC1                    ;ECC LOW ORDER TO BE GENERATED
0557 022160 005037 027130              CLR     00GECC2                    ;ECC HIGH ORDER TO BE GENERATED
0558 022164 005037 027144              CLR     00DATENV                   ;CLEAR DATA ENVELOPE CLOCK COUNT
0559 022170 005037 027146              CLR     00ZCODE                    ;CLEAR LEADING ZEROS CLOCK COUNT
0560
0561
0562                                     ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
0563
0564 022174 012737 010000 031016      MOV     00FMT22,00CYL             ;16 BITS PER WORD
0565                                     ;CYLINDER 0, FORMAT 16 BITS
0566 022202 112737 000000 031021      MOV     00, 00SECTOR+1           ;TRACK 0
0567 022210 112737 000000 031020      MOV     00, 00SECTOR             ;SECTOR 0
0568 022216 012737 000000 031022      MOV     00, 00KEY1                ;KEY1=0
0569 022224 012737 000000 031024      MOV     00, 00KEY2                ;KEY2=0
0570 022232 012737 000400 031076      MOV     0256,, 00DAWORD           ;NO. OF DATA WORDS
0571 022240 005037 031026              CLR     00X                        ;THIS IS A READ COMMAND
0572 022244 004537 025646              JSR     R5,00CRC                  ;GO TO CALCULATE CRC
0573 022250 031016
0574 022252 032716
0575
0576
0577
0578                                     ;THIS IS TO INSERT ERROR
0579                                     ;THE DISK DATA IS IN LOCATION STARTING FROM 'DISK'
0580                                     ;THE POSITION OF THE ERROR CAN BE CHANGED BY CHANGING
0581                                     ;THIS MOVE
0581 022254 012737 152525 032736      MOV     0152525,00DISK+2          ;FORCE ERROR ON BIT NUMBER 32
0582 022262 012737 152525 033732      MOV     0152525,00DISK+<255,+2>;FORCE ERROR IN BIT 4096
0583 022270 012737 010040 022436      MOV     04120,,0000              ;INSERT POSITION REG.
0584
0585
0586                                     ;THESE ARE REGULAR SETUPS
0587 022276 004737 024470              JSR     PC,00CLDISK              ;SETUP GENERAL REGISTERS
0588 022302 012777 177374 157314      MOV     0-256,-4,,00RHWC         ;256, DATA & HEADER WORDS
0589 022310 012777 003130 157310      MOV     00REINTO,00RHBA          ;STARTING ADDRESS OF READ BUFFER
0590 022316 112746 000000              MOV     00, -(SP)                 ;IN LOWER BYTE GET SECTOR
0591 022322 112766 000000 000001      MOV     00, 1(SP)                 ;GET TRACK IN HIGHER BYTE
0592 022330 012677 157302              MOV     (SP)+, 00RHDS            ;TRACK/SECTOR IN RHDST
0593 022334 012777 010000 157300      MOV     00FMT22,00RHOF          ;16 BITS PER WORD
0594                                     ;ECC CORRECTION NOT INHIBIT
0595                                     ;BECAUSE ECC IS NOT GOING
0596                                     ;TO BE CHECKED
0597 022342 005077 157276              CLR     00RHCA                    ;CYLINDER 0
0598
0599 022346 004737 024524              JSR     PC, 00CHECKT            ;CHECK FOR DVA,RDY,MOL,DPR,DRY
0600
0601 022352 013711 002046              MOV     00REFOR,00R1             ;READ HEADER AND DATA=72
0602 022356 005037 001774              CLR     00ERFLG0                 ;CLEAR ERROR FLAG

```

```

0603 022362 004737 030706      JSR    PC,      00COMHD ;READ HEADER AND DATA
0604                                ;IF THERE ARE READ ERRORS THEN
0605                                ;ECC WILL NOT BE CHECKED
0606
0607
0608                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
0609                                ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
0610                                ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
0611                                ;SYNC BYTE HAVE GONE BY AND SYNCs WERE CORRECTLY
0612                                ;DETECTED
0613                                ;HEADER AND DATA ARE TO BE CHECKED,
0614                                ;IN CHECKING READ DATA THE WRITE FROM BUFFER
0615                                ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
0616                                ;COMPARISONS ARE MADE
0617
0618 022366 005737 001774      TST    00ERFLG8          ;ANY ERRORS ALREADY THERE
0619
(1) 022372 001106      BNE    TST33    ;BRANCH IF YES
(1)
0620 022374 004737 024170      JSR    PC,00PUTREG      ;SAVE REGISTERS
0621 022400 022737 100000 001704  CMP    0DCK,00ER1      ;ONLY DATA CHECK ERROR SHOULD BE SET
0622 022406 001401          BEQ    68              ;BRANCH IF YES
0623 022410 104032          ERROR  32              ;32 BIT ECC REGISTER SHOULD BE NON
0624                                ;ZERO
0625                                ;ONLY 11 OF THE 32 BITS CAN BE SEEN
0626                                ;IN THE PATTERN REGISTER
0627                                ;DCK SHOULD BE SET IN RHER1
0628 022412 013746 027126          68:  MOV    00GECC1,-(SP)  ;GET PATTERN REGISTER
0629 022416 042716 174000          BIC    0174000,(SP)    ;KEEP ONLY 11 BITS
0630 022422 022637 001734          CMP    (SP)+,00EC2    ;COMPARE PATTERN REGISTER
0631 022426 001401          BEQ    78              ;BRANCH IF GOOD
0632 022430 104032          ERROR  32              ;11 BITS OF THE 32 BIT ECC REGISTER INCORRECT
0633
0634 022432 004037 027550          78:  JSR    R0,00ECORR   ;GO TO ECC CORRECTION PROCESS
0635 022436 000000          00:  ,WORD              ;EXPECTED POSITION REG, WHEN CORRECTION
0636                                ;IS COMPLETE
0637
0638
0639
0640 022440 004737 024170      JSR    PC,00PUTREG      ;SAVE REGISTERS
0641 022444 022737 100100 001704  CMP    0DCK|ECH,00ER1 ;WITH ERRORS INSERTED IN BIT POSITION 32
0642                                ;AND 4096 HARD ERROR BIT SHOULD SET
0643                                ;BRANCH IF GOOD
0644 022452 001401          BEQ    98              ;BRANCH IF GOOD
0645 022454 104036          ERROR  36              ;WITH ERROR INSERTED IN BIT POSITION 21 THRU
0646                                ;32 NCE SHOULD SET
0647
0648
0649
0650 022456 004737 024706          98:  JSR    PC,00CHECKE   ;CHECK DVA,DRY,RDY,DPR
0651 022462 012700 002064          MOV    0WRFROM,R0     ;GETTING READY TO FILL EXPECTED DATA
0652 022466 012720 010000          MOV    00IFMT22,(R0)+ ;CYLINDER 0
0653 022472 112746 000000          MOVB   00,      -(SP)  ;IN LOWER BYTE GET SECTOR
0654 022476 112766 000000 000001  MOVB   00,      1(SP)  ;GET TRACK IN HIGHER BYTE

```

```

0655 022504 012620      MOV      (SP)+, (R0)+ ;GET TRACK/SECTOR IN BUFFER
0656 022506 012720 000000  MOV      00, (R0)+ ;KEY1 IN BUFFER
0657 022512 012720 000000  MOV      00, (R0)+ ;KEY2 IN BUFFER
0658 022516 012701 000400  MOV      0256,, R1 ;DATA WORD COUNTER
0659 022522 012702 052525  MOV      052525, R2 ;DATA
0660 022526 010220      MOV      R2, (R0)+ ;DATA INTO BUFFER
0661 022530 005301      DEC      R1 ;COUNT
0662 022532 001375      BNE     30 ;BRANCH IF 256 NOT DONE
0663
0664 ;ONLY GOOD DATA HAS BEEN PUT IN 'WRFROM'
0665 ;NOW THE INSERTED ERROR WILL BE PUT IN
0666 022534 012737 152525 002076  MOV      0152525,00WRFROM+<5*2> ;INSERTED ERROR IN BIT 32
0667 022542 012737 152525 003072  MOV      0152525,00WRFROM+<259,*2> ;INSERT ERROR IN BIT 4096
0668
0669
0670
0671 022550 005037 001774      CLR     00ERFLG ;CLEAR ERROR FLAG
0672 022554 004737 024170      JSR     PC,00PUTREG ;SAVE REGISTERS
0673
0674
0675 ;NOW READ DATA BUFFER WILL BE CHECKED
0676
0677 022560 004037 025342      JSR     R0,00COMPAR ;CHECK
0678 022564 002064      WRFROM ;GOOD BUFFER
0679 022566 003130      REINTO ;TEST BUFFER
0680 022570 000404      4+256. ;NUMBER OF WORDS CHECKED
0681 022572 022600      40 ;RETURN POINT FOR ERROR HEADER
0682 022574 022604      50 ;RETURN POINT FOR ERROR DATA
0683
0684 (1) 022576 022610      TST33 ;RETURN FOR GOOD COMPARISON
0685 (1)
0686 022600 104004      401  ERROR 4 ;READ NEXT ERROR
0687 022602 000207      RTS   PC ;RETURN TO "COMPAR"
0688 022604 104005      501  ERROR 5 ;WORD NOS 1 TO 4 ARE
0689 ;HEADER WORDS
0690 ;5 TO 260 ARE DATA WORDS
0691 ;RETURN TO "COMPAR"
0692
0693
0694
0695
0696
0697
0698
0706
0707
(3) ;*****
(4) ;*TEST 33 PROGRAM INTERRUPT
(4) ;*
(4) ;* PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
(4) ;* IN RHC81 AT THE SAME TIME
(4) ;* THIS SHOULD INTERRUPT THROUGH LOCATION 254
(4) ;* THE PROCESSOR PRIORITY IS SET TO 4

```

```
(3) ;
(2) 022610 000004 TST33: SCOPE
8708
8709
(1) 022612 012737 000033 004174 NOV 0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
8710 022620 012706 001000 NOV 0STACK,SP ;RESET STACK
8711 022624 004737 024470 JSR PC,00CLDISK ;CLEAR DISK
8712 022630 013700 001620 NOV 00RPVEC,R0 ;GET VECTOR ADDRESS
8713 022634 012720 022702 NOV 0RPTRP1,(R0)+ ;SET INTERRUPT VECTOR
8714 022640 012710 000340 NOV 0340,(R0) ;SET SERVICE ROUTINE PRIORITY
8715 022644 012767 000200 155124 NOV 0200,PS ;SET PROCESSOR PRIORITY
8716 022652 012711 000300 NOV 0RDY,IE,0R1 ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
8717 022656 013737 025024 001172 NOV 00TIMCNT,008TMP1 ;COUNTER
8718 022664 005337 001172 10: DEC 008TMP1 ;WAIT FOR INTERRUPT
8719 022670 001375 BNE 10 ;BRANCH IF NOT ZERO
8720 ;BEFORE THIS IS ZERO INTERRUPT SHOULD
8721 ;OCCUR
8722 022672 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
8723 022676 104021 ERROR 21 ;INTERRUPT DID NOT OCCUR
8724
8725
(1) 022700 000410 BR TST34 ;BRANCH TO NEXT TEST
(1)
8726
8727 022702 022626 RPTRP1: CMP (SP)+,(SP)+ ;RESTORE STACK
8728 022704 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
8729 022710 022737 004200 001702 CMP 0DVAIRDY,00CS1 ;IE SHOULD BE LOW
8730
(1) 022716 001401 BEQ TST34 ;BRANCH IF GOOD
(1)
8731 022720 104021 ERROR 21 ;INTERRUPT OCCURED BUT
8732 ;IE FAILED TO RESET
8733
8739
8740 ;
(3) ;*TEST 34 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
(4)
(4) ;* PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
(4) ;* IE AND RDY IS SET. THIS SHOULD NOT INTERRUPT
(3) ;
(2) 022722 000004 TST34: SCOPE
8741
8742
(1) 022724 012737 000034 004174 NOV 0TTNO,00TSTNM ;THIS SAVES TEST NUMBER
(1)
8743 022732 012706 001000 NOV 0STACK,SP ;RESET STACK
8744 022736 004737 024470 JSR PC,00CLDISK ;CLEAR DISK
8745 022742 013700 001620 NOV 00RPVEC,R0 ;GET VECTOR ADDRESS
8746 022746 012720 023006 NOV 0RPTRP2,(R0)+ ;SET INTERRUPT VECTOR
8747 022752 012710 000340 NOV 0340,(R0) ;SET SERVICE ROUTINE PRIORITY
8748 022756 012767 000240 155012 NOV 0240,PS ;SET PROCESSOR PRIORITY
8749 022764 012711 000300 NOV 0RDY,IE,0R1 ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
8750 022770 013737 025024 001172 NOV 00TIMCNT,008TMP1 ;COUNTER
```

```

0751 022776 005337 001172          101  DEC  000TMP1      ;WAIT FOR INTERRUPT
0752 023002 001375                    BNE  10          ;BRANCH IF NOT ZERO
0753                                     ;BEFORE THIS IS ZERO INTERRUPT SHOULD
0754                                     ;OCCUR
0755
(1) 023004 000404                    BR   TST35      ;NO INTERRUPT SO BRANCH
(1)
0756
0757 023006 022626                    RPTRP2: CMP    (SP)+,(SP)+      ;RESTORE STACK
0758 023010 004737 024170             JSR    PC,03PUTREG             ;SAVE REGISTERS
0759 023014 104021                    ERROR  21                ;INTERRUPT OCCURRED WITH
0760                                     ;PROCESSOR STATUS SAME
0761                                     ;AS DISK
0762
0763
0764
0765
0766
0775
(3)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 023016 000004
(1) 023020 012767 000001 156156
0776 023026 004737 024470
0777 023032 012767 000000 154736
0778 023040 104400 023046
(1) 023044 000425
(1)
(1) 023120
0779 023120 013746 001762          648:
0780 023124 104410
0781 023126 104400 023134
(1) 023132 000402
(1)
(1) 023140          650:
0782 023140 013746 001112
0783 023144 104410
0784 023146 005037 001112
0785 023152 005737 001770
0786 023156 001415
0787 023160 005067 155716
0788 023164 005237 001100
0789 023170 104400 023366
0790 023174 013746 001100
0791 023200 104410
0792 023202 104400 023403
0793 023206 000137 007036
0794 023212 005337 001764          38:  DEC  00NOUNITS      ;JUMP TEST 5
                                     ;NO. OF UNITS PRESENT DECREMENT

```

8795	023216	001413		BEQ	SEOP	;BRANCH IF ALL DRIVES COMPLETE
8796	023220	013700	001762	MOV	00UNIT,R0	;UNIT UNDER TEST
8797	023224	012701	001742	MOV	SUNITS,R1	;TABLE
8798	023230	022100		CMP	(R1)+,R0	;IS THIS UNIT JUST TESTED
8799	023232	001401		BEQ	20	;BRANCH IF YES
8800	023234	000775		BR	10	;BRANCH IF NO
8801	023236	011137	001762	MOV	(R1),00UNIT	;THIS IS NEXT UNIT
8802	023242	000137	007036	JMP	00TST5	;GO FOR NEXT TESTS,

```

0008
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 023246
(1) 023246 000004
(1) 023250 005067 155626
(1) 023254 005067 155724
(1) 023260 005267 155614
(1) 023264 042767 100000 155606
(1) 023272 005327
(1) 023274 000001
(1) 023276 003031
(1) 023300 012737
(1) 023302 000001
(1) 023304 023274
(1) 023306 104400 023366
(2) 023312 016746 155562
(2) 023316 104410
(1) 023320 104400 023403
(1) 023324 013700 000042
(1) 023330 001414
(1) 023332 022700 023352
(1) 023336 001004
(1) 023340 022760 177777 000002
(1) 023345 001001
(1) 023350 000005
(1) 023352 004710
(1) 023354 000240
(1) 023356 000240
(1) 023360 000240
(1) 023362 000137 005340
(1) 023366 005015 047105 020104
(1) 023374 040520 051523 021440
(1) 023402 000
(1) 023403 377 377 000

```

```

;*****
.SBTTL END OF PASS ROUTINE

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

SEOP:
      SCOPE
      CLR STSTNM ;;ZERO THE TEST NUMBER
      CLR STIMES ;;ZERO THE NUMBER OF ITERATIONS
      INC SPASS ;;INCREMENT THE PASS NUMBER
      BIC $100000,SPASS ;;DON'T ALLOW A NEG. NUMBER
      DEC (PC)+ ;;LOOP?

SEOPCT: .WORD 1
      BGT SDOAGN ;;YES
      MOV (PC)+,0(PC)+ ;;RESTORE COUNTER

SENDCT: .WORD 1
      SEOPCT
      TYPE ,SENDMG ;;TYPE "END PASS #"
      MOV SPASS,-(SP) ;;SAVE SPASS FOR TYPEOUT
      TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
      TYPE ,SENULL ;;TYPE A NULL CHARACTER
      SGET42: MOV @42,R0 ;;GET MONITOR ADDRESS
      BEQ SDOAGN ;;BRANCH IF NO MONITOR
      CMP $SENDAD,R0 ;;IS MONITOR ACT11?
      BNE SRESET ;;NO--BRANCH (IT'S XXDP)
      CMP #-1,2(R0) ;;YES--IS THIS THE LAST PASS?
      BNE SENDAD ;;NO--MAKE ANOTHER PASS

SRESET: RESET ;;CLEAR THE WORLD
SENDAD: JSR PC,(R0) ;;GO TO MONITOR
      NOP ;;SAVE ROOM
      NOP ;;FOR
      NOP ;;ACT11
      SDOAGN: JMP @TST1 ;;RETURN
      SENDMG: .ASCIZ <15><12>/END PASS #/

SENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING

```

```

0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021

```

```

.SBTTL SUBROUTINES

;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.

```

```

0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835 023406 000000
0836 023410
0837 023410 005067 154362
0838 023414 104400 023422
(1) 023420 000421
(1)
(1) 023464
0839 023464 013746 004174
0840 023470 104402
0841 023472 104400 023500
(1) 023476 000414
(1)
(1) 023530
0842 023530 013746 001110
0843 023534 104402
0844 023536 104400 001215
0845 023542 104400 023550
(1) 023546 000426
(1)
(1) 023624
0846 023624 104400 023632
(1) 023630 000420
(1)
(1) 023672
0847 023672 104400 023700
(1) 023676 000423
(1)
(1) 023746
0848 023746 104416
0849 023750 062716 000002
0850 023754 012637 001106
0851 023760 104400 023766
(1) 023764 000417
(1)
(1) 024024
0852 024024 104400 024032
(1) 024030 000441
(1)
(1) 024134
0853 024134 104416
0854 024136 012637 001110
  
```

```

;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
TYPE ,,+4 ;;TYPE ASCII STRING
BR 640 ;;GET OVER THE ASCII
;;,ASCII <15><12>/THE PROGRAM WAS IN TEST NUMBER /
640:
MOV 00TSTNM,-(SP) ;GET READY TO TYPE TEST
TYPOC ;NUMBER
TYPE ,,+4 ;;TYPE ASCII STRING
BR 650 ;;GET OVER THE ASCII
;;,ASCII <15><12>/THE LOOP BACK PC WAS /
650:
MOV 000LPERR,-(SP) ;GET READY TO TYPE LOOP BACK PC
TYPOC
TYPE ,8CRLF
TYPE ,,+4 ;;TYPE ASCII STRING
BR 660 ;;GET OVER THE ASCII
;;,ASCII <15><12>/SET LOOP ON ERROR OR LOOP ON TEST SWITCH/
660:
TYPE ,,+4 ;;TYPE ASCII STRING
BR 670 ;;GET OVER THE ASCII
;;,ASCII <15><12>/TYPE THE FIRST PC OF THE TEST/
670:
TYPE ,,+4 ;;TYPE ASCII STRING
BR 680 ;;GET OVER THE ASCII
;;,ASCII <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
680:
RDOCT
ADD 02,(SP) ;GET LPADR
MOV (SP)+,000LPADR
TYPE ,,+4 ;;TYPE ASCII STRING
BR 690 ;;GET OVER THE ASCII
;;,ASCII <15><12>/TYPE THE PC WHERE YOU WANT/
690:
TYPE ,,+4 ;;TYPE ASCII STRING
BR 700 ;;GET OVER THE ASCII
;;,ASCII <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN
700:
RDOCT
MOV (SP)+,000LPERR ;GET LPERR
  
```



```

0055 024142 013746 001106      MOV      006LPADR,-(SP)
0056                                ;THIS CLEARS UP GARBAGE
0057 024146 005037 031132      CLR      00NOSYNC
0058 024152 005037 002012      CLR      00TSECC
0059                                ;CLEAR FLAG FOR HEADER ERROR COMMANDS
0060                                ;CLEAR FLAG FOR ECC TEST
0061                                ;WHEN =177777 IT IS AN ECC TEST
0062                                ;WHEN =0 IT IS NOT AN ECC TEST
0062 024156 005037 027132      CLR      00TSECCG
0063                                ;EVEN IN AN ECC TEST EVERY CLOCK
0064                                ;IS NOT TO GENERATE ECC
0065                                ;IF =177777 GENERATE ECC
0066 024162 005037 002014      CLR      00TESDTE
0067 024166 000002      RTI
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0900
0901
0902
0903
  
```

```

.SBTTL SAVE REGISTERS ROUTINE
;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:

```

(2) 024170 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
(2) 024172 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
(2) 024174 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
0081 024176 012700 001624      MOV      0RHWC,R0      ;STARTING ADDRESS OF REG
0082 024202 012701 001674      MOV      0WC,R1        ;STARTING ADDRESS OF WERE SAVED
0083 024206 012702 000023      MOV      0RHCC=RHWC+2/2,R2 ;NUMBER OF REG, INTO R2
0084 024212 013021      100: MOV      0(R0)+,(R1)+  ;SAVE HARDWARE REG,
0085 024214 005302      DEC      R2
0086 024216 001375      BNE     100
0087 024220 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
(2) 024222 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
(2) 024224 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
0088 024226 000207      RTS     PC
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0900
0901
0902
0903
  
```

```

.SBTTL FLOAT 1 AND 0
;FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
;ABSOLUTE ADDRESS OF REG, UNDER TEST IS IN R4
  
```

```

0098 024230 000000      MASK: 0      ;BITS UNDER TEST
0099 024232 000000      LERR: 0      ;ERROR HLT ADDRESS
0900 024234 000000      REGADR: 0
0901
0902 024236 012567 177766      BITST: MOV      (R5)+, MASK ;FETCH DATA MASK
0903 024242 012504      MOV      (R5)+, R4      ;GET ADDRESS OF REG, UNDER TEST
  
```

```

0904 024244 010467 177764      MOV      R4,      REGADR
0905 024250 010567 177756      MOV      R5,      LERR      ;GET ERROR RETURN ADDR.
0906 024254 062705 000004      ADD      04,      R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
0907 024260 012703 000001      MOV      01,      R3      ;INITIALIZE DATA PATTERN
0908 024264 004767 000016      BLT1:    JSR      PC,      BLT2  ;OUTPUT FLOATING ZERO
0909 024270 004767 000012      JSR      PC,      BLT2  ;OUTPUT FLOATING ONE
0910 024274 000241                CLC
0911 024276 006103                ROL      R3      ;SHIFT PATTERN
0912 024300 005703                TST      R3
0913 024302 001370                BNE      BLT1    ;BRANCH IF NOT COMPLETE
0914 024304 000205                RTS      R5      ;RETURN TO TEST
0915 024306 005103      BLT2:    COM      R3      ;COMPLEMENT PATTERN
0916 024310 012737 024316 024450      MOV      0BLT3, 00LAD  ;SET SCOPE LOOP
0917 024316 010337 001124      BLT3:    MOV      R3,00SGDDAT ;STORE GOOD DATA
0918 024322 005137 024230      COM      00MASK      ;AND MASK WITH PATTERN
0919 024326 043737 024230 001124      BIC      00MASK, 00SGDDAT ;CLEAR THE REST
0920 024334 005137 024230      COM      00MASK      ;RESTORE MASK
0921 024340 013714 001124      MOV      00SGDDAT,(R4) ;OUTPUT TO REGISTER
0922 024344 011437 001126      MOV      (R4),00SBDDAT ;INPUT FROM REGISTER
0923 024350 005137 024230      COM      00MASK
0924 024354 043737 024230 001126      BIC      00MASK,00SBDDAT ;AND MASK OUT RECEIVED DATA
0925 024362 005137 024230      COM      00MASK      ;RESTORE MASK
0926 024366 023737 001124 001126      CMP      00SGDDAT,00SBDDAT ;IS DATA CORRECT
0927 024374 001403                BEQ      10      ;BRANCH IF GOOD
0928 024376 004777 177630      JSR      PC,      0LERR  ;GO TO REPORT ERROR
0929 024402 104420                SCOP1    ;LOCAL SCOPE LOOP
0930 024404 000207      10:     RTS      PC
0931                .SBTTL  CLEAR MEMORY ROUTINE
0932
0933
0934
0935                ; THIS CLEARS ANY BLOCK OF MEMORY
0936                ; FILLING IT WITH ANY DATA
0937
0938                ;
0939                ; CALL
0940                ; JSR      R0,CLAREA
0941                ; X      ;STARTING ADDRESS OF BLOCK
0942                ; Y      ;
0943                ; Z      ;DATA TO BE FILLED
0944                ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
0945                ;R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
0946                ;R3 WILL HAVE DATA TO BE FILLED
0947                ;TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED
0947 024406      CLAREA:
(2) 024406 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
(2) 024410 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
(2) 024412 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
0948 024414 012001      MOV      (R0)+,R1      ;FROM
0949 024416 012002      MOV      (R0)+,R2      ;TO
0950 024420 012003      MOV      (R0)+,R3      ;DATA
0951 024422 160102      SUB      R1,R2      ;NO. OF LOCATIONS MINUS TWO
0952 024424 062702 000002      ADD      02,R2      ;GET TWICE NO OF LOCATIONS
0953 024430 010321      10:     MOV      R3,(R1)+
0954 024432 005302      DEC      R2

```

8955	024434	005302	DEC	R2	
8956	024436	001374	BNE	18	;BRANCH IF NOT COMPLETE
8957	024440	012603	MOV	(SP)+,R3	;POP STACK INTO R3
(2)	024442	012602	MOV	(SP)+,R2	;POP STACK INTO R2
(2)	024444	012601	MOV	(SP)+,R1	;POP STACK INTO R1
8958	024446	000200	RTS	R0	;RETURN

```

0960
0961 024450 000000          LAD:      .SBTTL LOCAL TRAPS
0962
0963 024452 032737 001000 177570 T,SCOPI BIT    0SW09, 00SWR
0964 024460 001402          BEQ      10
0965 024462 013716 024450          MOV     00LAD, (SP)
0966 024466 000002          10:     RTI
0967
0968          ;EXAMPLE OF THE USE OF THE ABOVE
0969          ;THIS WILL LOOP BETWEEN X: AND SCOPI PROVIDED THERE IS NO "NEWST"
0970          ;MOV     SX,      00LAD
0971          ;X:      ---      ---
0972          ;          ---      ---
0973          ;          ---      ---
0974          ;          SCOPI
0975
0976          .SBTTL CLEAD DISK ROUTINE
0977
0978          CLDISK: MOV     00RHCS1,      R1      ;R1 WILL BE CONTROL AND STATUS1
0979 024470 013701 001632          MOV     00RHCS2,      R2      ;R2 WILL BE CONTROL AND STATUS2
0980 024474 013702 001630          MOV     00RHDS1,      R3      ;R3 WILL BE DISK STATUS REGISTER1
0981 024500 013703 001654          MOV     00RHER1,      R4      ;R4 WILL BE ERROR REGISTER #1
0982 024504 013704 001634
0983
0984 024510 012712 000040          MOV     0CLR,0R2          ;CLEAR ALL REG.
0985 024514 013712 001762          MOV     00UNIT,0R2      ;REINSTATE UNIT NO.
0986 024520 005011          CLR     0R1              ;CLEAR FUNCTION BITS
0987 024522 000207          RTS     PC
0988
0989          .SBTTL CHECK DISK STATUS ROUTINE
0990
0991          ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
0992          ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
0993
0994          CHECKT: MOV     (SP),00PCJSR      ;SAVE PC OF JSR+4
0995          SUB     04,00PCJSR      ;GET PC OF JSR
0996 024524 011637 002002          JSR     PC,00PUTREG      ;SAVE REGISTERS
0997 024530 162737 000004 002002          CMP     0DVA,00CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
0998 024536 004737 024170          BEQ     30              ;AND BE READY
0999 024542 022737 004200 001702          BEQ     30              ;BRANCH IF GOOD
9000
9001 024550 001423          BIT     0DVA, 00CS1      ;BAD SO TEST DEVICE AVAILABLE
9002 024552 032737 004000 001702          BNE     10              ;BRANCH IF DVA THERE
9003 024560 001004          MOV     R1,000BDADR      ;ADDRESS OF BAD REGISTER (RHCS1)
9004 024562 010137 001122          ERROR  26              ;RHCS1 DID NOT HAVE DEVICE
9005 024566 104026          ;AVAILABLE RIGHT AT THE START
9006
9007 024570 000413          BR      30              ;BRANCH TO NEXT COMPARE
9008 024572 032737 000200 001702 10:    BIT     0RDY, 00CS1      ;TEST READY
9009 024600 001003          BNE     26              ;IF RDY THERE BRANCH
9010 024602 010137 001122          MOV     R1,000BDADR      ;ADDRESS OF BAD REGISTER (RHCS1)
9011 024606 104026          ERROR  26              ;RHCS1 DID NOT HAVE READY
9012
9013 024610 000403          20:    BR      30              ;RIGHT AT THE START
          ;BRANCH TO NEXT COMPARE
  
```

```

9014 024612 010137 001122      MOV      R1,000BDADR      ;ADDRESS OF BAD REGISTER (RHCS1)
9015 024616 104026      ERROR    26              ;RHCS1 HAD SOME BITS OTHER
9016                                     ;THAN DVA AND RDY SET
9017                                     ;ALL OTHER BITS SHOULD BE 0
9018 024620 013746 001724      30:     MOV      00DS1,-(SP)      ;GET RHDS1
9019 024624 042716 001100      BIC      0VVIPROG,(SP)    ;CLEAR VV AND PROGRAMABLE BIT
9020 024630 022726 000600      CMP      0DPR|DRY,(SP)+,RHDS1 SHOULD HAVE THESE SET
9021 024634 001423      BEQ      70              ;BRANCH IF GOOD
9022 024636 032737 000400 001724 40:     BIT      0DPR, 00DS1      ;TEST DRIVE PRESENT
9023 024644 001004      BNE      50              ;IF MOL WAS THERE SO BRANCH
9024 024646 010337 001122      MOV      R3,000BDADR      ;ADDRESS OF BAD REGISTER (RHDS1)
9025 024652 104026      ERROR    26              ;RHDS1 DOES NOT HAVE DPR
9026 024654 000413      BR       70              ;BRANCH OUT
9027 024656 032737 000200 001724 50:     BIT      0DRY, 00DS1      ;TEST DRIVE READY
9028 024664 001004      BNE      60              ;IF DPR WAS THERE SO BRANCH
9029 024666 010337 001122      MOV      R3,000BDADR      ;ADDRESS OF BAD REGISTER (RHDS1)
9030 024672 104026      ERROR    26              ;RHDS1 DOES NOT HAVE DRY
9031 024674 000403      BR       70              ;BRANCH OUT
9032 024676 010337 001122      60:     MOV      R3,000BDADR      ;ADDRESS OF BAD REGISTER (RHDS1)
9033 024702 104026      ERROR    26              ;RHDS1 HAS SOME BITS OTHER
9034                                     ;THAN MOL, DRY, DPR, SET
9035                                     ;ALL OTHER BITS SHOULD BE 0
9036 024704 000207      70:     RTS      PC              ;RETURN TO TEST NO.
9037
9038
9039
9040
9041                                     ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
9042                                     ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
9043 024706 011637 002002      CHECKE: MOV      (SP),00PCJSR      ;SAVE PC OF JSR+4
9044 024712 162737 000004 002002      SUB      04,00PCJSR      ;GET PC OF JSR
9045 024720 004737 024170      JSR      PC,00PUTREG      ;SAVE REGISTERS
9046 024724 032737 000200 001702      BIT      0RDY,00CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
9047                                     ;AND BE READY
9048 024732 001004      BNE      10              ;BRANCH IF GOOD
9049 024734 010137 001122      MOV      R1,000BDADR      ;FAILING REGISTER
9050 024740 104026      ERROR    26              ;RHCS1 IS IN ERROR
9051                                     ;DOES NOT HAVE DVA, RDY
9052 024742 000427      BR       40              ;BRANCH
9053 024744 032737 004000 001702 10:     BIT      0DVA,00CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
9054                                     ;AND BE READY
9055 024752 001004      BNE      20              ;BRANCH IF GOOD
9056 024754 010137 001122      MOV      R1,000BDADR      ;FAILING REGISTER
9057 024760 104026      ERROR    26              ;RHCS1 IS IN ERROR
9058                                     ;DOES NOT HAVE DVA, RDY
9059 024762 000417      BR       40              ;BRANCH OUT
9060 024764 032737 000200 001724 20:     BIT      0DRY,00DS1      ;RHDS1 SHOULD HAVE DPR,DRY
9061 024772 001004      BNE      30              ;BRANCH IF THERE
9062 024774 010337 001122      MOV      R3,000BDADR      ;FAILING REGISTER RHDS1
9063 025000 104026      ERROR    26              ;RHDS1 DOES NOT HAVE DPR,DRY
9064 025002 000407      BR       40              ;BRANCH OUT
9065 025004 032737 000400 001724 30:     BIT      0DPR,00DS1      ;RHDS1 SHOULD HAVE DPR,DRY
9066 025012 001003      BNE      40              ;BRANCH IF THERE
9067 025014 010337 001122      MOV      R3,000BDADR      ;FAILING REGISTER RHDS1

```

```

9068 025020 104026          ERROR 26          ;RHDS1 DOES NOT HAVE DPR,DRY
9069 025022 000207          401  RT6      PC
9070
9071
9072
9073
9074
9075
9076
9077          ;          WAIT LOOP
9078          ; ONE LOOP OR ONE COUNT = 5.15 MICROSEC WITH BIPOLAR MEMORY (MIN)
9079          ; ONE LOOP OR ONE COUNT = 11.86 MICROSEC WITH CORE (MIN)
9080          ; WITH CORE ERROR IS INDICATED AFTER ABOUT 650 MILLISEC (MIN)
9080 025024 177777          TIMCNT: 177777          ;WAITING COUNT
9081 025026 010046          WAIT,T: MOV      R0,-(SP)          ;SAVE R0
9082 025030 016600 000002          MOV      2(SP),R0          ;GET ADDRESS OF REG, ADDRESS
9083 025034 010037 001176          MOV      R0,000TMP3          ;WAT PC+2 IN 0TMP3
9084 025040 162737 000002 001176          SUB      #2,000TMP3          ;WAT PC FOR TYPEOUT
9085 025046 012037 001170          MOV      (R0)+,000TMP0          ;WAIT REGISTER ADDRESS
9086 025052 012037 001172          MOV      (R0)+,000TMP1          ;WAIT ON BIT
9087 025056 010066 000002          MOV      R0,2(SP)          ;RESTORE RETURN ON STACK
9088 025062 012600          MOV      (SP)+,R0          ;RESTORE R0
9089 025064 013737 025024 001174          MOV      00TIMCNT,000TMP2          ;TEMPORARY COUNT
9090 025072 033777 001172 154070 101          BIT      000TMP1,00TMP0          ;IS REQUIRED BIT THERE?
9091 025100 001021          BNE      20          ;BRANCH IF YES
9092 025102 005337 001174          DEC      000TMP2          ;COUNT
9093 025106 001371          BNE      10          ;BRANCH IF NOT TIME UP
9094 025110 013737 025024 001174          MOV      00TIMCNT,000TMP2          ;TEMPORARY COUNT
9095 025116 033777 001172 154044 301          BIT      000TMP1,00TMP0          ;IS REQUIRED BIT THERE?
9096 025124 001007          BNE      20          ;BRANCH IF YES
9097 025126 005337 001174          DEC      000TMP2          ;COUNT
9098 025132 001371          BNE      30          ;BRANCH IF NOT TIME UP
9099 025134 017737 154030 001126          MOV      00TMP0,000BDDAT          ;REGISTER CONTENTS
9100 025142 104016          ERROR 16          ;WAITED ON BIT FAILED TO SET
9101 025144 000002          201  RTI
9102          ; CALL FOR THE ABOVE WAITLOOP IS
9103          ;
9104          ; MOV      0A,00X0          ;A CONTAINS REGISTER ADDRESS
9105          ; - - -          ;HENCE X0 WILL HAVE ABSOLUTE REG. ADR.
9106          ; - - -
9107          ; - - -
9108          ; WAT
9109          ; X0: 0          ;ABSOLUTE REG. ADDRESS UNDER WAIT
9110          ; ,WORD 0          ;BIT WAITED FOR
9111          ;
9112          ;
9113          ;
9114          ;
9115          ; THIS IS A SUBROUTINE TO SAVE REGISTERS
9116          ; IN THE REGISTER TABLE TO ANY LOCATION
9117          ; .SBTTL SAVE ROUTINE
9118          ; THE CALL IS
9119          ; JSR      R0,00SAVER
9120          ; FROM
9121          ; TO
  
```

```

9122                                     ;NUMBER OF WORDS SAVED
9123
9124 025146 SAVER1
(2) 025146 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
(2) 025150 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
(2) 025152 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
9125 025154 012001 MOV (R0)+,R1 ;FROM
9126 025156 012002 MOV (R0)+,R2 ;TO
9127 025160 012003 MOV (R0)+,R3 ;NUMBER
9128 025162 013122 101 MOV 0(R1)+,(R2)+ ;SAVE REGISTER CONTENTS
9129 025164 005303 DEC R3 ;COUNT
9130 025166 001375 BNE 10 ;BRANCH IF NOT DONE
9131 025170 012603 MOV (SP)+,R3 ;POP STACK INTO R3
(2) 025172 012602 MOV (SP)+,R2 ;POP STACK INTO R2
(2) 025174 012601 MOV (SP)+,R1 ;POP STACK INTO R1
9132 025176 000200 RTS R0
9133
9134
9135
9136
9137 .SBTTL WRITE CHECK ROUTINE
9138 ;THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
9139 ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
9140
9141 ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
9142 025200 012737 010000 031016 WRCHHD: MOV 0FMT22,00CYL ;CYLINDER 0 FORMAT 16 BIT WORDS
9143 025206 112737 000001 031021 MOV 01,00SECOTR+1 ;TRACK=1
9144 025214 112737 000001 031020 MOV 01,00SECOTR ;SECTOR=1
9145 025222 005037 031022 CLR 00KEY1 ;KEY1=0
9146 025226 005037 031024 CLR 00KEY2 ;KEY2=0
9147 025232 012767 000044 003636 MOV 036,,DAWORD ;NO OF DATA WORDS
9148 025240 005037 031026 CLR 00X ;THIS IS A READ OPERATION
9149 025244 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
9150 025250 031016 CYL
9151 025252 032716 WCRC
9152
9153 ;THESE ARE REGULAR SETUPS
9154
9155 025254 004737 024470 JSR PC,00CLDISK ;SET UP GENERAL REGISTERS
9156 ;AND CLEAR DISK REGISTERS
9157 025260 012777 177730 154336 MOV 0-40,,0RHWC ;36 DATA WORDS 4 HEADER WORDS
9158 025266 012777 003130 154332 MOV 0REINTO,0RHBA ;STARTING ADDRESS OF READ BUFFER
9159 025274 112746 000001 MOV 01,-(SP) ;SECTOR=1
9160 025300 112766 000001 000001 MOV 01,1(SP) ;TRACK=1 IN UPPER BYTE
9161 025306 012677 154324 MOV (SP)+,0RHDS1 ;TRACK=1, SECTOR=1 IN RHDST
9162 025312 012777 014000 154322 MOV 0FMT22!ECI,0RHOF ;16 BIT WORDS
9163 ;ECC CORRECTION INHIBIT BECAUSE
9164 ;ECC LOGIC IS NOT CHECKED YET
9165 025320 005077 154320 CLR 0RHCA ;CYLINDER=0
9166 025324 004737 024524 JSR PC,00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
9167 025330 013711 002036 MOV 00WRCHDT,0R1 ;WRITE CHECK HEADER AND DATA=02
9168 ;INTO RHC01
9169 025334 004737 030706 JSR PC,00COMHD ;WRITE CHECK HEADER AND DATA
9170 ;SAME AS READ HEADER AND DATA

```

```

9171
9172 025340 000207                                RTS      PC                                ;RETURN TO WRITE CHECK TEST
9173                                .SBTTL COMPARE ROUTINE
9174                                ;THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
9175                                ;R1 HAS GOOD DATA BUFFER ADDRESS
9176                                ;R2 HAS TEST DATA BUFFER ADDRESS
9177                                ;TMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER
9178                                ;TMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA
9179                                ;R3 HAS NUMBER OF WORDS TO BE COMPARED
9180                                ;R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
9181
9182 025342                                COMPARE:
(2) 025342 010146                                MOV      R1,-(SP)                                ;;PUSH R1 ON STACK
(2) 025344 010246                                MOV      R2,-(SP)                                ;;PUSH R2 ON STACK
(2) 025346 010346                                MOV      R3,-(SP)                                ;;PUSH R3 ON STACK
(2) 025350 010446                                MOV      R4,-(SP)                                ;;PUSH R4 ON STACK
(2) 025352 010546                                MOV      R5,-(SP)                                ;;PUSH R5 ON STACK
9183 025354 012001                                MOV      (R0)+,R1                                ;ADDRESS OF GOOD DATA BUFFER
9184 025356 012002                                MOV      (R0)+,R2                                ;ADDRESS OF TEST DATA BUFFER
9185 025360 012003                                MOV      (R0)+,R3                                ;NO OF WORDS TO BE COMPARED
9186 025362 012067 153602                                MOV      (R0)+,TMP0                                ;RETURN ON ERROR TO PRINT HEADER
9187 025366 012067 153600                                MOV      (R0)+,TMP1                                ;RETURN ON ERROR TO PRINT DATA
9188 025372 011000                                MOV      (R0),R0                                ;RETURN ON NO ERROR
9189 025374 010304                                MOV      R3,R4                                ;NO OF WORDS TO BE COMPARED
9190 025376 005204                                INC      R4
9191 025400 010437 031136                                10: MOV      R4,00ERWORD                                ;FOR ERROR WORD NO
9192 025404 022122                                CMP      (R1)+,(R2)+                                ;COMPARE GOOD WITH TEST DATA
9193 025406 001426                                BEQ      30                                ;BRANCH IF GOOD
9194
9195 025410 014137 001124                                MOV      -(R1),000GDDAT                                ;GOOD DATA
9196 025414 014237 001126                                MOV      -(R2),000BDDAT                                ;BAD DATA
9197 025420 160337 031136                                SUB      R3,00ERWORD                                ;ERROR WORD NO,
9198 025424 005737 001774                                TST      00ERFLG0                                ;ANY ERRORS ALREAY THERE
9199 025430 001003                                BNE      20                                ;BRANCH IF YES
9200 025432 004777 153532                                JSR      PC,00TMP0                                ;RETURN TO PRINT HEADER
9201 025436 000402                                BR      50                                ;BRANCH TO AVOID PRINTING NEXT ERROR
9202 025440 004777 153526                                20: JSR      PC,00TMP1                                ;RETURN TO PRINT DATA
9203 025444 022122                                50: CMP      (R1)+,(R2)+                                ;UNDO -(R1) AND -(R2) FOR ERRORS
9204 025446 013746 177570                                MOV      00SWR,-(SP)                                ;GET SWITCH SETTING
9205 025452 042716 177177                                BIC      0°C600,(SP)                                ;KEEP ONLY SWITCH 7 AND 8
9206 025456 022726 000200                                CMP      0SW07,(SP)+                                ;IS 7 SET AND 8 RESET
9207 025462 001402                                BEQ      40                                ;BRANCH OUT IF YES
9208 025464 005303                                30: DEC      R3                                ;COUNT
9209 025466 001344                                BNE      10                                ;BRANCH IF ALL NOT DEVICE
9210 025470                                40:
(2) 025470 012605                                MOV      (SP)+,R5                                ;;POP STACK INTO R5
(2) 025472 012604                                MOV      (SP)+,R4                                ;;POP STACK INTO R4
(2) 025474 012603                                MOV      (SP)+,R3                                ;;POP STACK INTO R3
(2) 025476 012602                                MOV      (SP)+,R2                                ;;POP STACK INTO R2
(2) 025500 012601                                MOV      (SP)+,R1                                ;;POP STACK INTO R1
9211 025502 000200                                RTS      R0                                ;RETURN TO MAIN PROGRAM
9212
9213
9214

```



```

9215
9216
9217
9218
9219
9220 025504 012737 010000 031016 WRCHDA: MOV 0FMT22,00CYL ;THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
9221 025512 112737 000001 031021 MOV 01,00SECOTR+1 ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
;CYLINDER 0 FORMAT 16 BIT WORDS
9222 025520 112737 000001 031020 MOV 01,00SECOTR ;TRACK=1
;SECTOR=1
9223 025526 005037 031022 CLR 00KEY1 ;KEY1=0
9224 025532 005037 031024 CLR 00KEY2 ;KEY2=0
9225 025536 012737 000040 031076 MOV 032,,00DAWORD ;NO OF DATA WORDS
9226 025544 005037 031026 CLR 00X ;THIS IS A READ OPERATION
9227
9228 025550 004537 025646 JSR R5,00CRC ;GO TO CALCULATE CRC
9229 025554 031016 CYL
9230 025556 032716 WCRC
9231
9232 ;THESE ARE REGULAR SETUPS
9233
9234 025560 004737 024470 JSR PC,00CLDISK ;SET UP GENERAL REGISTERS
;AND CLEAR DISK REGISTERS
9235
9236
9237 025564 012777 177740 154032 MOV 0-32,,0RHWC ;36 DATA WORDS 4 HEADER WORDS
9238 025572 012777 003130 154026 MOV 0REINT0,0RHBA ;STARTING ADDRESS OF READ BUFFER
9239 025600 112746 000001 MOV 01,-(SP) ;SECTOR=1
9240 025604 112766 000001 000001 MOV 01,1(SP) ;TRACK=1 IN UPPER BYTE
9241 025612 012677 154020 MOV (SP)+,0RHDBT ;TRACK=1, SECTOR=1 IN RHDST
9242 025616 012777 014000 154016 MOV 0FMT22!ECI,0RHOF ;16 BIT WORDS
9243 ;ECC CORRECTION INHIBIT BECAUSE
9244 ;ECC LOGIC IS NOT CHECKED YET
9245 025624 005077 154014 CLR 0RHCA ;CYLINDER=0
9246 025630 004737 024524 JSR PC,00CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
9247 025634 013711 002034 MOV 00WRCHK,0R1 ;WRITE CHECK DATA=00 INTO RMC81
9248 025640 004737 030706 JSR PC,00CONHD ;WRITE CHECK HEADER AND DATA
;SAME AS READ HEADER AND DATA
9249
9250
9251 025644 000207 RTS PC ;RETURN TO WRITE CHECK TEST
9252
9253
9254 ;SBTTL CRC GENERATION ROUTINE
9255 ;THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
9256 ;HEADER WORDS AND STORE THEM IN "WCRC" AND "GCRC"
9257 ;R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
9258 ;R2 - THIS HAS BIT POSITION 2 VALUE C
9259 ;R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
9260 ;R4 - THIS HAS BIT POSITION 18 VALUE E
9261 ;STMP0 - NUMBER OF WORDS
9262 ;STMP2 - NUMBER OF BITS PER WORD = 16
9263 ;STMP3 - TEMPORARY REG.
9264 ;STMP4 - TEMPORARY REG TO TRANSFER CARRY
9265 ;STMP5 - THIS HAS DATA BIT VALUE D
9266
9267 ;FETCH DATA BIT D
9268 ;B = D XOR 16
;C = B XOR 2

```

```

9269                                     ;E = B XOR 15
9270                                     ;ROTATE RIGHT ONE POSITION
9271                                     ;B GOES TO POSITION 1
9272                                     ;C GOES TO POSITION 3
9273                                     ;E GOES TO POSITION 16
9274                                     ;REPET 64 TIMES
9275                                     ;CALL    JSR      R5,00CRC
9276                                     ;X      ;FIRST LOCATION AT
9277                                     ;Y      ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE
9278
9279    025646                              CRC:
(2)    025646    010046                    MOV      R0,-(SP)                    ;;PUSH R0 ON STACK
9280    025650    012500                    MOV      (R5)+,R0                   ;GET POINTER TO CYL NO.
9281    025652    010146                    MOV      R1,-(SP)                   ;;PUSH R1 ON STACK
(2)    025654    010246                    MOV      R2,-(SP)                   ;;PUSH R2 ON STACK
(2)    025656    010346                    MOV      R3,-(SP)                   ;;PUSH R3 ON STACK
(2)    025660    010446                    MOV      R4,-(SP)                   ;;PUSH R4 ON STACK
9282    025662    005001                    CLR      R1                        ;CLEAR WORKING LOCATION
9283    025664    005037    001202           CLR      008TMP5
9284    025670    012737    000004    001170           MOV      04,008TMP0               ;WORD COUNT
9285    025676    012037    001176           MOV      (R0)+,008TMP3             ;TEMPORARY WORD STORAGE
9286    025702    012767    000020    153264           MOV      016,,8TMP2               ;BIT COUNT
9287    025710    013737    001176    001200           MOV      008TMP3,008TMP4         ;TEMPORARY WORD STORAGE
9288    025716    006037    001176           ROR      008TMP3                   ;GET LSB INTO "C"
9289    025722    006037    001202           ROR      008TMP5                   ;GET ABOVE "C" INTO 8TMP5
9290    025726    032701    000001           BIT      0BIT0,R1                 ;IS POSITION 15 HIGH
9291    025732    001403                    BEQ      16                        ;BRANCH IF POSITION 16 LOW
9292    025734    012703    100000           MOV      0BIT15,R3                ;GET POSITION 16
9293    025740    000401                    BR      20
9294    025742    005003                    CLR      R3                        ;GET POSITION 16
9295    025744    063703    001202           ADD      008TMP5,R3                ;XOR POSITION 16 WITH D
9296                                                                                    ;TO GIVE B
9297    025750    032701    040000           BIT      0BIT14,R1                ;IS POSITION 2 HIGH
9298    025754    001403                    BEQ      36                        ;BRANCH IF POSITION 2 LOW
9299    025756    012702    100000           MOV      0BIT15,R2                ;GET POSITION 2
9300    025762    000401                    BR      40
9301    025764    005002                    CLR      R2                        ;GET POSITION 2
9302    025766    060302                    ADD      R3,R2                     ;XOR B WITH POSITION 2
9303                                                                                    ;TO GIVE C
9304    025770    032701    000002           BIT      0BIT1,R1                 ;IS POSITION 15 HIGH
9305    025774    001403                    BEQ      56                        ;BRANCH IF POSITION 15 LOW
9306    025776    012704    100000           MOV      0BIT15,R4                ;GET POSITION 15
9307    026002    000401                    BR      60
9308    026004    005004                    CLR      R4                        ;GET POSITION 15
9309    026006    060304                    ADD      R3,R4                     ;XOR POSITION 15 WITH B
9310                                                                                    ;TO GIVE E
9311    026010    006037    001200           ROR      008TMP4                   ;GET LSB INTO "C"
9312    026014    006001                    ROR      R1                        ;GET ABOVE C INTO R1
9313    026016    005703                    TST      R3                        ;TEST B
9314    026020    100403                    BMI      78                        ;BRANCH IF B=1
9315    026022    042701    100000           BIC      0BIT15,R1                ;SET B IN POSITION 1
9316    026026    000402                    BR      100
9317    026030    052701    100000           BIS      0BIT15,R1                ;SET B IN POSITION 1
9318    026034    005702                    TST      R2                        ;TEST C

```

9319	026036	100403		BMI	110		;BRANCH IF C=1
9320	026040	042701	020000	BIC	0BIT13,R1		;GET C IN POSITION 3
9321	026044	000402		BR	120		
9322	026046	052701	020000	BIS	0BIT13,R1	1101	;GET C IN POSITION 3
9323	026052	005704		TST	R4	1201	;TEST E
9324	026054	100403		BMI	130		;BRANCH IF E=1
9325	026056	042701	000001	BIC	0BIT0,R1		;GET E IN POSITION 16
9326	026062	000402		BR	140		
9327	026064	052701	000001	BIS	0BIT0,R1	1301	;GET E IN POSITION 16
9328	026070	005337	001174	DEC	000TMP2	1401	;BIT COUNTER
9329	026074	001310		BNE	150		;BRANCH IF 16 NOT DONE
9330	026076	005337	001170	DEC	000TMP0		;WORD COUNTER
9331	026102	001275		BNE	160		;BRANCH IF 4 NOT DONE
9332	026104	010135		MOV	R1,0(R5)+		;PUT CRC WHERE DESIRED
9333	026106	012604		MOV	(SP)+,R4		;POP STACK INTO R4
(2)	026110	012603		MOV	(SP)+,R3		;POP STACK INTO R3
(2)	026112	012602		MOV	(SP)+,R2		;POP STACK INTO R2
(2)	026114	012601		MOV	(SP)+,R1		;POP STACK INTO R1
(2)	026116	012600		MOV	(SP)+,R0		;POP STACK INTO R0
9334	026120	000205		RTS	R5		
9335							
9336							
9337							
9338							
9339							
9340							
9341							
9342							
9343							
9344							
9345							
9346							
9347							
9348							
9349	026122						
(2)	026122	010046					
(2)	026124	010146					
(2)	026126	010246					
9350	026130	012700	177400				
9351	026134	012701	000400				
9352	026140	012702	032734				
9353	026144	010022					
9354	026146	005301					
9355	026150	001375					
9356	026152	012701	000021				
9357							
9358	026156	005022					
9359							
9360	026160	005301					
9361	026162	001375					
9362							
9363							
9364							
9365	026164	012737	010000 031016				

```

;THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
;CYLINDER 0 (16 BITS PER WORD)
;TRACK 1, SECTOR 1
;KEY1 1
;KEY2 1
;CRC THROUGH THE JBR R5,00CRC
;256 WORDS OF 177400

```

```

;CALL JBR PC,00SETDSK

```

```

SETDSK:

```

```

MOV R0,-(SP) ;PUSH R0 ON STACK
MOV R1,-(SP) ;PUSH R1 ON STACK
MOV R2,-(SP) ;PUSH R2 ON STACK
MOV 0177400,R0 ;DATA IN THE DISK
MOV 0256,,R1 ;COUNTER
MOV 0DISK,R2 ;START OF SIMULATOR DISK
MOV R0,(R2)+ ;MOVE IN DATA
DEC R1 ;COUNT FOR 256
BNE 10 ;BRANCH IF 256 NOT COMPLETE
MOV 017,,R1 ;2 ECC WORDS, 1 DATA GAP
;14 TOLERANCE GAP
CLR (R2)+ ;CLEAR ECC,DATA GAP AND
;TOLERANCE GAP
DEC R1 ;COUNT
BNE 20 ;BRANCH IF NOT COMPLETE

```

```

;NOW SET UP FOR DISKLESS USE

```

```

MOV 0FMT22,00CYL ;CYLINDER 0 (16 BIT WORDS)

```

9366	026172	112737	000001	031021	MCVB	#1,0#SECOTR+1	;TRACK=1
9367	026200	112737	000001	031020	MOV	#1,0#SECOTR	;SECTOR=1
9368	026206	012737	000001	031022	MOV	#1,0#KEY1	;KEY1=1
9369	026214	012737	000001	031024	MOV	#1,0#KEY2	;KEY2=1
9370	026222	016737	152152	031076	MOV	256,,0#DAWORD	;NO, OF DATA WORDS
9371	026230	004537	025646		JSR	R5,0#CRC	;GO TO CALCULATE CRC
9372	026234	031016			CYL		;FIRST CRC WORD
9373	026236	032716			WCRC		;PUT CALCULATED CRC
9374	026240	012602			MOV	(SP)+,R2	;POP STACK INTO R2
(2)	026242	012601			MOV	(SP)+,R1	;POP STACK INTO R1
(2)	026244	012600			MOV	(SP)+,R0	;POP STACK INTO R0
9375	026246	000207			RTS	PC	
9376							
9377							
9378							
9379							
9380							
9381							
9382							
9383							
9384							
9385							
9386							
9387							
9388							
9389							
9390	026250	010037	002002				
9391	026254	162737	000004	002002	HCCRCE:	MOV R0,0#PCJSR	;SAVE PC OF JSR+4
9392	026262	004737	024470			SUB #4,0#PCJSR	;GET PC OF JSR
9393						JSR PC,0#CLDISK	;INIT AND SETUP GENERAL REG.
(1)							
(1)	026266	004767	176232			JSR PC,CHECKT	;CHECK DVA, RDY, DPR, DRY
(1)							
9394	026272	011037	001202			MOV (R0),0#STMP5	;SAVE COMMAND
9395	026276	012011				MOV (R0)+,0R1	;COMMAND
9396	026300	012077	153340			MOV (R0)+,0RHCA	;CYLINDER
9397	026304	112046				MOV (R0)+,-(SP)	;SECTOR
9398	026306	105720				TSTB (R0)+	;UP DATE R0
9399	026310	112066	000001			MOV (R0)+,1(SP)	;TRACK
9400	026314	105720				TSTB (R0)+	;UPDATE R0
9401	026316	012677	153314			MOV (SP)+,0RHDST	;TRACK SECTOR
9402	026322	012077	153276			MOV (R0)+,0RHWC	;NO, OF DATA WORDS +4 HEADER
9403							;IF A READ HEADER AND DATA
9404	026326	012077	153274			MOV (R0)+,0RHBA	;STARTING ADDRESS OF BUFFER
9405	026332	012037	031026			MOV (R0)+,00X	;X=0 READ HEADER AND DATA
9406							;X=1 WRITE DATA
9407	026336	012777	014000	153276		MOV #FMT22IECI,0RHOF	;16 BITS PER WORD
9408							;ECC CORRECTION INHIBIT
9409	026344	005037	001774			CLR #0ERFLG8	;CLEAR ERROR FLAG
9410	026350	004737	030706			JSR PC,0#COMHD	;COMMAND
9411							
9412							
9413							
9414							

;THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
 ;(BIT 07) AND CRC ERROR (BIT 00)
 ;CALL JSR R0,0#HCCRCE
 ;
 ; COM ;COMMAND=READ HEADER AND DATA
 ; ;WRITE DATA
 ;
 ; C ;CYLINDER
 ; S ;SECTOR
 ; T ;TRACK
 ; -N. ;WORD COUNT
 ; B ;RHBA BUFFER START
 ; X ;1=WRITE DATA 0=READ
 ; H ;H=1 HEADER CHECK, H=0 CRC CHECK

;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
 ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND


```

9469 026454 000207          RTS      PC          ;RETURN TO COMPARISON SUBROUTINE
9470 026456 005720          TST      (R0)+       ;IS THIS A HCRC ON HCE CHECK?
9471 026460 001442          BEQ      68          ;BRANCH IF HCRC
9472 026462 022737 000072 001202  CMP      072,000TMP5 ;IS THIS A READ COMMAND
9473 026470 001417          BEQ      118         ;BRANCH IF YES
9474 026472 017737 153136 001126  MOV      0RHER1,000BDDAT ;TEST DATA
9475 026500 022737 000200 001126  CMP      0HCE,000BDDAT ;ONLY HEADER COMPARE BIT?
9476                                ;SHOULD BE SET
9477 026506 001470          BEQ      78          ;BRANCH IF GOOD
9478 026510 013737 001634 024234  MOV      00RHER1,00REGADR ;REGISTER ADDRESS RHER1
9479 026516 012737 000200 001124  MOV      0HCE,000GDDAT ;GOOD DATA
9480 026524 104027          ERROR   27          ;AFTER AN ERROR ON THE
9481                                ;HEADER ONLY HCE SHOULD
9482 026526 000460          BR       78          ;BE SET
9483 026530          118:
9484 026530 017737 153100 001126  MOV      0RHER1,000BDDAT ;TEST DATA
9485 026536 022737 100200 001126  CMP      0DCK;HCE,000BDDAT ;ONLY HEADER COMPARE BIT?
9486                                ;SHOULD BE SET
9487                                ;DCK IS SET BECAUSE ECC IS NOT READ
9488 026544 001451          BEQ      78          ;BRANCH IF GOOD
9489 026546 013737 001634 024234  MOV      00RHER1,00REGADR ;REGISTER ADDRESS RHER1
9490 026554 012737 100200 001124  MOV      0DCK;HCE,000GDDAT ;GOOD DATA
9491 026562 104027          ERROR   27          ;AFTER AN ERROR ON THE
9492                                ;HEADER ONLY HCE SHOULD
9493 026564 000441          BR       78          ;BE SET
9494 026566 022737 000072 001202 68:  CMP      072,000TMP5 ;IS THIS A READ COMMAND?
9495 026574 001417          BEQ      128         ;BRANCH IF A READ
9496 026576 017737 153032 001126  MOV      0RHER1,000BDDAT ;TEST DATA
9497 026604 022737 000400 001126  CMP      0HCRC,000BDDAT ;ONLY CRC ERROR SHOULD BE THERE
9498 026612 001426          BEQ      78          ;BRANCH IF GOOD
9499 026614 013737 001634 024234  MOV      00RHER1,00REGADR ;REG. ADDR = RHER1
9500 026622 012737 000400 001124  MOV      0HCRC,000GDDAT ;GOOD DATA
9501 026630 104027          ERROR   27          ;AFTER A CRC ERROR ONLY CRC
9502                                ;SHOULD BE SET
9503 026632 000416          BR       78          ;BRANCH OUT
9504 026634 017737 152774 001126 128:  MOV      0RHER1,000BDDAT ;TEST DATA
9505 026642 022737 100400 001126  CMP      0DCK;HCRC,000BDDAT;HCRC AND DCK SHOULD BE SET
9506                                ;DCK IS SET BECAUSE ECC IS NOT READ
9507                                ;BRANCH IF GOOD
9508 026650 001407          BEQ      78          ;BRANCH IF GOOD
9509 026652 012737 100400 001124  MOV      0DCK;HCRC,000GDDAT;GOOD DATA
9510 026660 013737 001634 024234  MOV      00RHER1,00REGADR;FAILING REGISTER RHER1
9511 026666 104027          ERROR   27          ;AFTER A CRC ERROR ON A READ
9512                                ;DCK AND HCRC SHOULD BE SET
9513                                ;DCK IS SET BECAUSE ECC IS NOT READ
9514 026670 000200          78:  RTS      R0          ;RETURN TO MAIN TEST
9515
9516
9517
9518                                ;THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
9519                                ;A WRITE HEADER AND DATA COMMAND
9520                                ;IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
9521                                ;BUT COMES OUT AFTER ONE SECTOR
9522                                ;THE COMMAND OS JSR PC,00MIDDLE

```

```

9523                                     ;BAI IS SET
9524
9525 026672                                MIDDLE:
  (2) 026672 010046                       MOV    R0,-(SP)           ;;PUSH R0 ON STACK
  (2) 026674 010146                       MOV    R1,-(SP)           ;;PUSH R1 ON STACK
9526 026676 013777 002042 152726         MOV    00WRIFOR,0RHCS1  ;WRITE HEADER AND DATA=62
9527                                     ;IN RHCS1
9528 026704 012777 177766 152712         MOV    0-10,,0RHWC       ;10 WORDS
9529 026712 012777 002064 152706         MOV    0WRFROM,0RHBA    ;BUS ADDRESS=WRFROM
9530 026720 012777 000010 152710         MOV    010,0RHDS1       ;DESIRED TRACK=0 SECTOR=10
9531 026726 052777 000010 152674         BIS    0BAI,0RHCS2      ;BUS ADDRESS INCREMENT INHIBIT
9532 026734 012777 010000 152700         MOV    0FMT22,0RHOF     ;FORMAT 16 BIT WORDS
9533 026742 005077 152676                 CLR    0RHCA            ;CYLINDER=0
9534 026746 012737 000001 026774         MOV    01,00HID        ;SECTOR IS SET TO 1 SO THAT
9535                                     ;WE CAN GET OUT AT THE
9536                                     ;MIDDLE OF AN OPERATION
9537                                     ;LOOKING FOR SECTOR 10
9538 026754 012777 000001 152670         MOV    0DND,0RHMR       ;SET DIAGNOSTIC MODE
9539 026762 052777 000001 152642         BIS    0GO,0RHCS1       ;GO TO RHCS1 WITH 62
9540 026770 004137 035032                 JSR    R1,00SEARCH
9541 026774 000000                                MID: ;SECTOR
9542 026776 012601                                MOV    (SP)+,R1          ;POP STACK INTO R1
  (2) 027000 012600                                MOV    (SP)+,R0          ;POP STACK INTO R0
9543 027002 000207                                RTS    PC
9544
9545
9546
9547                                     ;BTTL JAM CURRENT CYLINDER ROUTINE
9548 ;THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
9549 ;THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
9550 ;WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
9551 ;
9552 ;CALL IS
9553 ;JSR R0,00MAKECYL
9554 ;XC ;DESIRED VALUE OF CURRENT CYLINDER
9555
9556
9557 027004                                MAKECYL:
  (2) 027004 010546                       MOV    R5,-(SP)           ;;PUSH R5 ON STACK
9558 027006 010037 002002 002002         MOV    R0,00PCJSR       ;PC OF JSR+4
9559 027012 162737 000004 002002         SUB    04,00PCJSR       ;SAVE PC OF JSR
9560 027020 012005 (R0)+,R5           ;GETTING READY TO FILL DESIRED CYLINDER
9561 027022 010577 152616                       MOV    R5,0RHCA         ;FILL DESIRED CYLINDER REGISTER
9562 027026 005077 152604                       CLR    0RHDS1           ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
9563 027032 013777 002050 152572         MOV    00SEECOM,0RHCS1 ;FILL SEEK COMMAND
9564 027040 012777 000001 152604         MOV    0DND,0RHMR       ;SET DIAGNOSTIC MODE
9565 027046 052777 000001 152556         BIS    0GO,0RHCS1       ;GO TO SEEK
9566 027054 000240                                NOP                       ;ALLOW TIME FOR SEEK TO HANG UP
9567 027056 000240                                NOP                       ;ALLOW TIME FOR SEEK TO HANG UP
9568 027060 000240                                NOP                       ;ALLOW TIME FOR SEEK TO HANG UP
9569 027062 000240                                NOP                       ;ALLOW TIME FOR SEEK TO HANG UP
9570 027064 004737 024470                       JSR    PC,00CLDISK      ;GIVE INIT
9571 027070 017737 152574 001126         MOV    0RHCC,000BDDAT  ;TEST DATA
9572 027076 020537 001126                       CMP    R5,000BDDAT      ;COMPARE CURRENT CYLINDER

```

```

9573 027102 001406                    BEQ      18                    ;BRANCH IF GOOD
9574 027104 010537 001124            MOV      R5,008GDDAT        ;GOOD VALUE OF RHCC
9575 027110 013737 001670 024234    MOV      00RHCC,00REGADR   ;FAILING REGISTER ADDRESS
9576 027116 104030                    ERROR    30                    ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
9577                                    ;REGISTER AFTER A SEEK AND AN INIT
9578 027120                    181                    ;
(2) 027120 012605                    MOV      (SP)+,R5           ;;POP STACK INTO R5
9579 027122 000200                    RTS      R0

```

.SBTTL ECC GENERARION AND COMPARISON ROUTINE

;THIS SUBROUTINE GENERATES AND TESTS ECC
 ;CALL JSR PC,ECTest

9594	100000	PIE1	=100000
9595	040000	PIE2	=40000
9596	020000	PIE3	=20000
9597	010000	PIE4	=10000
9598	004000	PIE5	=4000
9599	002000	PIE6	=2000
9600	001000	PIE7	=1000
9601	000400	PIE8	=400
9602	000200	PIE9	=200
9603	000100	PIE10	=100
9604	000040	PIE11	=40
9605	000020	PIE12	=20
9606	000010	PIE13	=10
9607	000004	PIE14	=4
9608	000002	PIE15	=2
9609	000001	PIE16	=1
9610	100000	PIE17	=100000
9611	040000	PIE18	=40000
9612	020000	PIE19	=20000
9613	010000	PIE20	=10000
9614	004000	PIE21	=4000
9615	002000	PIE22	=2000
9616	001000	PIE23	=1000
9617	000400	PIE24	=400
9618	000200	PIE25	=200
9619	000100	PIE26	=100
9620	000040	PIE27	=40
9621	000020	PIE28	=20
9622	000010	PIE29	=10
9623	000004	PIE30	=4
9624	000002	PIE31	=2
9625	000001	PIE32	=1


```

9626
9627 027124 000000      ECDATA: 0          ;DATA BIT FOR ECC
9628                                     ;IF ALL ONES THEN CURRENT BIT IS A ONE
9629                                     ;IF ZERO THEN CURRENT BIT IS A ZERO
9630
9631 027126 000000      GECC1: 0          ;LOW ORDER ECC WORD TO BE GENERATED HERE
9632                                     ;=R1
9633
9634 027130 000000      GECC2: 0          ;HIGH ORDER ECC WORD TO BE GENERATED HERE
9635                                     ;=R2
9636
9637 027132 000000      TSECCG: 0         ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
9638                                     ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT
9639
9640 027134 113713      NCODE: 38859.     ;N-CODE WORD
9641 027136 000000      NCOUNT: 0        ;TEMPORARY N CODE
9642 027140 000000      POSITI: 0         ;POSITION REGISTER
9643 027142 010041      HARDER: 4129.    ;HARD ERROR COUNT
9644                                     ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
9645                                     ;DONE ONE STAGE LATER SO 4129
9646 027144 000000      DATENV: 0        ;DATA ENVELOPE FOR TYPE OUT
9647                                     ;MAX FOR WRITE IS 4096
9648                                     ;MAX FOR READ IS 4128
9649 027146 000000      ZCODE: 0         ;LEADING ZEROS ENVELOPE FOR TYPE OUT
9650                                     ;THIS IS SHUT OFF WHEN POSITION COUNTER
9651                                     ;IN ENABLED
9652                                     ;MAX COUNT IS 38859
9653
9654
9655
9656 027150 000000      HADTMP: 0         ;TEMPORARY HARD ERROR COUNT
9657 027152 000000      P3: 0
9658 027154 000000      P12: 0
9659 027156 000000      P22: 0
9660 027160 000000      P24: 0
9661
9662
9663
9664
9665
9666 027162      ECTEST:
(2) 027162 010046      MOV R0,-(SP)        ;PUSH R0 ON STACK
(2) 027164 010146      MOV R1,-(SP)        ;PUSH R1 ON STACK
(2) 027166 010246      MOV R2,-(SP)        ;PUSH R2 ON STACK
(2) 027170 010346      MOV R3,-(SP)        ;PUSH R3 ON STACK
(2) 027172 010446      MOV R4,-(SP)        ;PUSH R4 ON STACK
(2) 027174 010546      MOV R5,-(SP)        ;PUSH R5 ON STACK
9667 027176 013701 027126  MOV 00GECC1,R1      ;ECC1 WORD
9668 027202 013702 027130  MOV 00GECC2,R2      ;ECC2 WORD
9669 027206 005737 027124  TST 00ECDATA        ;IS CURRENT BIT A ONE
9670 027212 001406      BEQ 20              ;BRANCH IF CURRENT DATA D=0
9671                                     ;IF CARRY IS NOT ZERO THEN D=1
9672                                     ;INVERT X32 TO GIVE R0
9673 027214 010103      10: MOV R1,R3

```

9674	027216	052703	177776		BIS	0°CPIE32,R3	
9675	027222	005103			COM	R3	
9676	027224	010300			MOV	R3,R0	
9677	027226	000404			BR	38	
9678							
9679							
9680							
9681	027230	010103		201			
9682	027232	042703	177776		MOV	R1,R3	
9683	027236	010300			BIC	0°CPIE32,R3	
9684					MOV	R3,R0	
9685	027240	000241		301			
9686	027242	006000			CLC		
9687	027244	006000			ROR	R0	
9688	027246	005700			ROR	R0	
9689	027250	001462			TST	R0	
9690					BEQ	100	
9691							
9692	027252	010203					
9693	027254	052703	137777				
9694	027260	005103			MOV	R2,R3	
9695	027262	010337	027152		BIS	0°CPIE2,R3	
9696	027266	006237	027152		COM	R3	
9697					MOV	R3,00P3	
9698					ASR	00P3	
9699							
9700							
9701	027272	010203					
9702	027274	052703	177737				
9703	027300	005103					
9704	027302	010337	027154				
9705	027306	006237	027154				
9706							
9707							
9708							
9709	027312	010103					
9710	027314	052703	173777				
9711	027320	005103					
9712	027322	010337	027156				
9713	027326	006237	027156				
9714							
9715							
9716							
9717	027332	010103					
9718	027334	052703	176777				
9719	027340	005103					
9720	027342	010337	027160				
9721	027346	006237	027160				
9722							
9723							
9724							
9725							
9726							
9727							

```

;IF CARRY IS ZERO THEN D=0
;X32 BECOMES R0
;BRANCH IF R0=0
;INVERT X2
;INVERT X11
;INVERT X21
;INVERT X23
;INVERT X24
;NOW THAT R0 FOR POSITION 1
;          P3 FOR POSITION 3
;          P12 FOR POSITION 12
;          P22 FOR POSITION 22
;          P24 FOR POSITION 24

```

```

9720                                     ;ARE KNOWN THE ROTATE WILL BE DONE AND
9729                                     ;THESE BITS JAMED IN
9730
9731 027352 006002                       ROR    R2
9732 027354 006001                       ROR    R1
9733 027356 053700 027152                BIS    00P3,R0
9734 027362 053700 027154                BIS    00P12,R0
9735 027366 042702 120020                BIC    0PIE1,PIE3,PIE12,R2
9736 027372 050002                       BIS    R0,R2
9737
9738 027374 005000                       CLR    R0
9739 027376 053700 027156                BIS    00P22,R0
9740 027402 053700 027160                BIS    00P24,R0
9741 027406 042701 002400                BIC    0PIE22,PIE24,R1
9742 027412 050001                       BIS    R0,R1
9743 027414 000404                       BR     120
9744
9745                                     ;THE PROGRAM COMES HERE IF R0=0
9746                                     ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
9747 027416 006002                       100:  ROR    R2
9748 027420 006001                       ROR    R1
9749 027422 042702 100000                BIC    0PIE1,R2
9750 027426 010137 027126                120:  MOV    R1,00GECC1 ;SAVE ECC1
9751 027432 010237 027130                MOV    R2,00GECC2 ;SAVE ECC2
9752 027436 005737 027132                TST    00TSECC    ;IS HARDWARE TO BE CHECKED
9753                                     ;IF =1777777 TEST HARDWARE
9754                                     ;IF = 0 DO NOT TEST HARDWARE
9755 027442 001432                       BEQ    140         ;BRANCH IF HARDWARE NOT TO BE CHECKED
9756
9757
9758                                     ;CHECK HARDWARE
9759 027444 032737 000400 177570          BIT    0SW0,00SWR ;IS SWITCH 0 SET
9760 027452 001005                       BNE    150         ;BRANCH IF SW0 IS SET
9761 027454 032737 000100 177570          BIT    0SW6,00SWR ;IS SWITCH 6 SET
9762 027462 001401                       BEQ    150         ;BRANCH IF SW6 IS NOT SET
9763 027464 000421                       BR     140         ;IF SWITCH 0 IS NOT SET AND
9764                                     ;SWITCH 6 IS SET THEN
9765                                     ;DO NOT DO COMPARES
9766 027466 010146                       150:  MOV    R1,-(SP) ;GOOD PATTERN REGISTER
9767 027470 042716 174000                BIC    0174000,(SP) ;GET ONLY PATTERN BITS
9768 027474 022677 152164                CMP    (SP)+,0RHEC2 ;COMPARE PATTERN REGISTER
9769 027500 001404                       BEQ    130         ;BRANCH IF GOOD
9770                                     ;TO SAVE TIME
9771 027502 004737 024170                JSR    PC,00PUTREG ;SAVE REGISTERS
9772 027506 104035                       ERROR  35         ;PATTERN REGISTER IN 11 BITS IN ERROR
9773 027510 000407                       BR     140         ;BRANCH OUT
9774 027512 023777 027140 152142 130:  CMP    00POSITI,0RHEC1 ;COMPARE POSITION REGISTER
9775 027520 001403                       BEQ    140         ;BRANCH IF GOOD
9776                                     ;TO SAVE TIME
9777 027522 004737 024170                JSR    PC,00PUTREG ;SAVE REGISTERS
9778 027526 104035                       ERROR  35         ;POSITION REGISTER IN ERROR
9779                                     ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
9780                                     ;PULSES FROM BEGINING OF COMMAND
9781                                     ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE

```

```

9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797 027530      1481
(2) 027530 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
(2) 027532 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
(2) 027534 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
(2) 027536 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
(2) 027540 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
(2) 027542 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
9798 027544 000207      RTS      PC
9799
9800
9801      ;THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
9802      ;FOR ERROR CORRECTION PROCESS
9803      ;CALL JSR, PC,00ECORR
9804      ; XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
9805
9806
9807
9808 027546 000000      ERPOS: 0      ;POSITION REG. WHEN CORRECTION IS COMPLETE
9809
9810
9811
9812 027550 010037 002002      ECORR: MOV      R0,00PCJSR      ;SAVE PC OF JSR + 4
9813 027554 162737 000004 002002      SUB      04,00PCJSR      ;SAVE PC OF JSR
9814 027562 012037 027546      MOV      (R0)+,00ERPOS      ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
9815 027566 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9816 027570 013701 001652      MOV      00RHMR,R1      ;MAINTENANCE REGISTER
9817 027574 012711 000001      MOV      00DMD,0R1      ;SET DIAGNOSTIC MODE BIT
9818 027600 005037 027124      CLR      00ECDATA      ;ECC DATA IS ZERO
9819
9820
9821
9822 027604 005737 027140      181      TST      00POSITI      ;IS SOFTWARE POSITION NON ZERO
9823 027610 001007      BNE      28      ;BRANCH IF N-CODE 8 COMPLETE
9824 027612 005337 027136      DEC      00NCOUNT      ;DECREMENT N-CODE
9825 027616 001001      BNE      68      ;BRANCH IF N-CODE IS NOT COMPLETE
9826 027620 000403      BR       28      ;BRANCH AS N-CODE IS COMPLETE
9827 027622 005237 027146      681      INC      00ZCODE      ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
9828 027626 000420      BR       38      ;BRANCH AS N-CODE IS NOT COMPLETE
9829

```

```

;
;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
;IN A READ THERE ARE 10040 OCTAL CLOCKS
;
;
;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
; GIVEN FOR THE LEADING ZEROS FIELD
;MAX COUNT IS 113713 OCTAL
;
;"GOOD POSITION" GIVES NUMBER OF CLOCKS
;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
;FIELD
;MAX COUNT IS 10040 OR 10041 OCTAL

```

```

9830 027630 005237 027140 281 INC 00POSITI ;INCREMENT SOFTWARE POSITION
9831 027634 023737 027546 027140 CMP 00ERPOS,00POSITI ;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
9832 027642 103012 30 BHS ;BRANCH IF MORE CLOCKS TO BE GIVEN
9833 027644 023737 027150 027140 CHF 00HADTMP,00POSITI ;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
9834 ;THAT IS HAVE 4120 MORE CLOCKS BEEN GIVEN
9835 027652 001415 BEQ 50 ;BRANCH IF YES
9836 027654 032711 000400 BIT 0ZER,0R1 ;CHECK ZERO DETECT BIT IN RHMR
9837 027660 001016 BNE 40 ;BRANCH IS ZER SET
9838 ;TO SAVE TIME
9839 027662 004737 024170 JSR PC,00PUTREG ;SAVE REGISTERS
9840 027666 104034 ERROR 34 ;ZERO DETECT BIT NOT HIGH
9841 ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS 0
9842
9843
9844 027670 052711 000002 301 BIS 0MCLK,0R1 ;SET CLOCK
9845 027674 042711 000002 BIC 0MCLK,0R1 ;CLEAR CLOCK
9846 027700 004737 027162 JSR PC,00ECTEST ;GO TO GENERATE AND TEST ECC
9847 027704 000737 BR 10 ;CONTINUE
9848
9849 ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
9850 ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL
9851 027706 052711 000002 501 BIS 0MCLK,0R1 ;SET CLOCK
9852 027712 042711 000002 BIC 0MCLK,0R1 ;CLEAR CLOCK
9853
9854 027716 401
(2) 027716 012601 MOV (SP)+,R1 ;POP STACK INTO R1
9855 027720 000200 RTS R0
9856
9857
9858
9859
9860
9861
9862 ;THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
9863 ;ON LOCATIONS "DISK+1000" AND "DISK+1002"
9864
9865
9866
9867 027722 FILLEC:
(2) 027722 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
(2) 027724 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
(2) 027726 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
(2) 027730 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
(2) 027732 010446 MOV R4,-(SP) ;PUSH R4 ON STACK
(2) 027734 010546 MOV R5,-(SP) ;PUSH R5 ON STACK
9868 027736 005037 027140 CLR 00POSITI ;CLEAR POSITION
9869 027742 005037 027126 CLR 00GECC1 ;CLEAR GECC1
9870 027746 005037 027130 CLR 00GECC2 ;CLEAR
9871 027752 012701 032734 MOV 0DISK,R1 ;POINTER TO DATA FOR ECC GENERATION
9872 027756 012702 000400 MOV 0256,,R2 ;COUNTER FOR NUMBER OF DATA WORDS
9873 027762 012703 000020 901 MOV 016,,R3 ;COUNTER FOR NUMBER OF BITS PER WORD
9874 027766 012104 MOV (R1)+,R4 ;DATA IN R4
9875 027770 006004 1001 ROR R4 ;GET ONE DATA BIT IN CARRY
9876 027772 103004 BCC 110 ;BRANCH IF DATA BIT IS ZERO

```

```

9877 027774 012737 177777 027124      MOV    0-1,00ECDATA      ;ECC DATA BIT IS A ONE
9878 030002 000402                    BR     128              ;BRANCH TO GENERATE ECC
9879 030004 005037 027124      118:  CLR    00ECDATA      ;ECC DATA BIT IS A ZERO
9880 030010 004737 027162      128:  JSR    PC,00ECTEST    ;GO TO GENERATE ECC
9881 030014 005303                    DEC    R3              ;DECREMENT BIT COUNT
9882 030016 001364                    BNE   108              ;BRANCH IF 16 BITS NOT DONE
9883 030020 005302                    DEC    R2              ;DECREMENT WORD COUNT
9884 030022 001357                    BNE   98              ;BRANCH IF 256 WORDS NOT DONE
9885 030024 013737 027126 033734    MOV    00GECC1,00DISK+<256,02>;INSERT ECC1 ON DISK
9886 030032 013737 027130 033736    MOV    00GECC2,00DISK+<257,02>;INSERT ECC2 ON DISK
9887 030040 012605                    MOV    (SP)+,R5        ;;POP STACK INTO R5
(2) 030042 012604                    MOV    (SP)+,R4        ;;POP STACK INTO R4
(2) 030044 012603                    MOV    (SP)+,R3        ;;POP STACK INTO R3
(2) 030046 012602                    MOV    (SP)+,R2        ;;POP STACK INTO R2
(2) 030050 012601                    MOV    (SP)+,R1        ;;POP STACK INTO R1
(2) 030052 012600                    MOV    (SP)+,R0        ;;POP STACK INTO R0
9888 030054 000207                    RTS     PC
  
```

```

;SBTTL RH BASE ADDRESS CHANGE ROUTINE
; THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
; ADDRESS FROM 176700 TO ANY TYPED VALUE
  
```

```

9898 030056 104400 030064      BASECH:
(1) 030056 104400 030064      TYPE    ,,+4          ;;TYPE ASCIZ STRING
(1) 030062 000424                    BR     648            ;GET OVER THE ASCIZ
(1)                                ;;,ASCIZ <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS/
648:
9899 030134 013746 001632      MOV    00RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
9900 030140 104402      TYP0C
9901 030142 104400 030150      TYPE    ,,+4          ;;TYPE ASCIZ STRING
(1) 030146 000425                    BR     658            ;GET OVER THE ASCIZ
(1)                                ;;,ASCIZ <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
658:
9902 030222 104416      RDOCT
9903 030224 012700 001622      MOV    00RHDB,R0      ;GET STARTING ADDRESS OF REGISTERS
9904 030230 012701 000024      MOV    020,,R1        ;NUMBER OF REGISTERS
9905 030234 042710 177700      18:   BIC    0"C77,(R0)     ;CLEAR OLD BASE
9906 030240 051620                    BIS    (SP),(R0)+     ;SET NEW BASE
9907 030242 005301                    DEC    R1              ;COUNT
9908 030244 001373                    BNE   18              ;BRANCH IF 20 NOT DONE
9909 030246 104400 030254      TYPE    ,,+4          ;;TYPE ASCIZ STRING
(1) 030252 000417                    BR     668            ;GET OVER THE ASCIZ
(1)                                ;;,ASCIZ <15><12>/PRESENT VECTOR ADDRESS IS /
668:
9910 030312 013746 001620      MOV    00RPVEC,-(SP)  ;GET READY TO TYPE OLD VECTOR ADDRESS
9911 030316 104402      TYP0C
9912 030320 104400 030326      TYPE    ,,+4          ;;TYPE ASCIZ STRING
(1) 030324 000437                    BR     678            ;GET OVER THE ASCIZ
(1)                                ;;,ASCIZ <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR"
(1) 030424                    678:
  
```

```

9913 030424 104416                    RDOCT
9914 030426 012637 001620            NOV      (SP)+,00RPVEC      ;SETUP VECTOR ADDRESS
9915 030432 104400 030440            TYPE     ,,+4                ;;TYPE ASCIZ STRING
(1) 030436 000421                    BR       688                ;;GET OVER THE ASCIZ
(1)                                    ;;ASCIZ      <15><12>/RESTART PROGRAM FROM 200 OR 210/
(1) 030502                            688:
9916 030502 104400 030510            TYPE     ,,+4                ;;TYPE ASCIZ STRING
(1) 030506 000414                    BR       698                ;;GET OVER THE ASCIZ
(1)                                    ;;ASCIZ      <15><12>/NEW BASE WILL REMAIN/
(1) 030540                            698:
9917 030540 013746 001632            NOV      00RHCS1,-(SP)
9918 030544 104402                    TYPOC
9919 030546 104400 030554            TYPE     ,,+4                ;;TYPE ASCIZ STRING
(1) 030552 000415                    BR       708                ;;GET OVER THE ASCIZ
(1)                                    ;;ASCIZ      <15><12>/NEW VECTOR WILL REMAIN /
(1) 030606                            708:
9920 030606 013746 001620            NOV      00RPVEC,-(SP)
9921 030612 104402                    TYPOC
9922 030614 104400 030622            TYPE     ,,+4                ;;TYPE ASCIZ STRING
(1) 030620 000416                    BR       718                ;;GET OVER THE ASCIZ
(1)                                    ;;ASCIZ      <15><12>/UNTIL PROGRAM IS RELOADED/
(1) 030656                            718:
9923 030656 000000                    HALT

```

9924
9925
9926
9927
9928
9929
9930
9931
9932

;THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
 ;THIS LOOPS HERE FOR EVER
 ;TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
 ;WITH WHAT IS REALY THERE

```

9933 030660 000000                    ERUNIT: 0                    ;UNIT UNDER MANUAL TEST
9934 030662 004737 024470            ERSTART:JSR      PC,00CLDISK      ;SET GENERAL REG,
9935 030666 013712 030660                       NOV      00ERUNIT,0R2      ;SELECT UNIT
9936 030672 005714                    10:            TST      0R4                ;TEST RHER1
9937 030674 032712 010000                       BIT      0NED,0R2            ;TEST NED
9938 030700 001401                               BEQ      20                ;BRANCH IF GOOD
9939 030702 000773                               BR       10                ;NED NOT SET
9940 030704 000772                    20:            BR       10                ;NED SET

```

9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995

```
                  .SBTTL    DISK SIMULATION  
;*****  
;*****  
;IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING  
;WCLY=WITH CYLINDER TO BE ON DISK  
;WSECTR=WITH SECTOR AND TRACK TO BE ON DISK  
;WKEY1= WITH KEY1 TO BE ON DISK  
;WKEY2= WITH KEY2 TO BE ON DISK  
;FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK  
;THE COMMAND THEN IS    JSR PC,COMHD  
;  
;  
;  
;IN A WRITE DATA COMMAND FILL THE FOLLOWING  
;CYL=WITH CYLINDER TO BE FOUND ON DISK  
;SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK  
;KEY1= WITH KEY1 TO BE FOUND ON DISK  
;KEY2= WITH KEY2 TO BE FOUND ON DISK  
;X= 1    MUST BE ONE  
;NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN  
;THE COMMAND THEN IS JSR PC,COMHD  
;  
;  
;  
;IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING  
;CYL= WITH CYLINDER TO BE FOUND ON DISK  
;SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK  
;KEY1= WITH KEY1 TO BE FOUND ON DISK  
;KEY2=WITH KEY2 TO BE FOUND ON DISK  
;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK  
;;X=0 MUST BE ZERO  
;THE COMMAND THEN IS JSR PC,COMHD  
;  
;  
;  
;IN A READ DATA COMMAND FILL THE FOLLOWING  
;CYL= WITH CYLINDER TO BE FOUND ON DISK  
;SECTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK  
;KEY1= WITH KEY1 TO BE FOUND ON DISK  
;KEY2=WITH KEY2 TO BE FOUND ON DISK  
;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK  
;;X=0 MUST BE ZERO  
;THE COMMAND THEN IS JSR PC,COMHD
```


9996

1

9998
 9999
 10000
 10001
 10002
 10003
 10004
 10005
 10006
 10007 030706 011637 002002
 10008 030712 162737 000004 002002
 10009 030720 010046
 (2) 030722 010146
 (2) 030724 010246
 (2) 030726 010346
 (2) 030730 010446
 (2) 030732 010546
 10010 030734 012777 000001 150710
 10011 030742 052777 000001 150662
 10012 030750 016746 000044
 10013 030754 042716 177740
 10014 030760 012637 030770
 10015 030764 004137 035032
 10016 030770 000000
 10017 030772 012701 000240
 10018 030776 010137 031030
 10019 031002 010137 031032
 10020 031006 010137 031034
 10021 031012 004137 031140
 10022 031016 000000
 10023 031020 000000
 10024 031022 000000
 10025 031024 000000
 10026 031026 000000
 10027
 10028 031030 000240
 10029
 10030
 10031
 10032
 10033
 10034
 10035
 10036
 10037
 10038 031032 000240
 10039
 10040
 10041
 10042
 10043
 10044
 10045
 10046

```

;WRITE DATA COMMAND
;OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA
COMHD:  MOV      (SP),00PCJSR      ;SAVE PC OF JSR + 4
        SUB      #4,00PCJSR      ;SAVE PC OF JSR
        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      @DMD,@RHRM      ;SET DIAGNOSTIC MODE
        BIS      @GO,@RHC&1      ;GO
        MOV      SECOTR,-(SP)    ;GET DESIRED SECTOR/TRACK
        BIC      @177740,(SP)    ;MAKE ONLY SECTOR
        MOV      (SP)+,@TRK      ;SAVE SECTOR
        JSR      R1,@SEARCH      ;DO SEARCH SECTOR
        TRK:    ,WORD            0
        MOV      @+NOP,R1        ;GOING TO MOVE NOPS
        MOV      R1,@SSYN        ;NOP INTO SSYN
        MOV      R1,@HEDGAP      ;NOP INTO HEDGAP
        MOV      R1,@HEDSYN      ;NOP INTO HEDSYN
        JSR      R1,@RDHEAD
        CYL:    ,WORD            0 ;CYLINDER ADDRESS
        SECOTR: ,WORD            0 ;SECTOR/TRACK ADDRESS
        KEY1:   ,WORD            0 ;KEY1 WORD
        KEY2:   ,WORD            0 ;KEY2 WORD
        X:      ,WORD            0 ;X=1 WRITE COMMAND
        ;X=0 READ COMMAND
        ;IF "ERROR 2" INSERTED BY RDHEAD
        ;SUBROUTINE THEN THE FIRST SYNC,
        ;IS NOT DETECTED, NO BAD DATA
        ;IS GIVEN BECAUSE SYNC=144000
        ;CANNOT BE READ, WORD NO
        ;IS "1" BECAUSE THIS IS THE FIRST
        ;WORD TESTED
        HEDGAP: NOP
        ;IF "ERROR 3" INSERTED BY
        ;RDHEAD SUBROUTINE THEN THE
        ;HEADER GAP 0'S WERE NOT
        ;WRITTEN RIGHT,
        ;IF "WORD NO" CONTAINS SAY
        ;3(0) THEN IT IS THE THIRD
        ;WORD OF A 5 WORD HEADER
        ;GAP THAT IS WRONG
        ;"BAD DATA" CONTAINS WHAT IS
  
```

10047
 10048
 10049
 10050
 10051
 10052
 10053
 10054
 10055
 10056
 10057
 10058
 10059
 10060
 10061
 10062
 10063
 10064
 10065
 10066
 10067
 10068
 10069
 10070
 10071
 10072
 10073
 10074
 10075
 10076
 10077
 (2)
 (2)
 (2)
 (2)
 (2)
 (2)
 (2)
 10078
 10079
 10080
 10081
 10082
 10083
 10084
 10085
 10086
 10087
 10088
 10089
 10090
 10091
 10092
 10093
 10094

031034 000240
 031036 005737 001774
 031042 001017
 031044 005737 031026
 031050 001410
 031052 005737 031132
 031056 001011
 031060 004137 032404
 031064 000000
 031066 000000
 031070 000404
 031072 004137 035306
 031076 000000
 031100 000000
 031102 012605
 031104 012604
 031106 012603
 031110 012602
 031112 012601
 031114 012600
 031116 000207
 031120 014400
 031122 000000
 031124 000000
 031126 000000
 031130 000000

HEDSYN: NOP

TST 00ERFLG0
 BNE OUT
 TST 00X
 BEQ DAREAD
 TST 00NOSYNC
 BNE OUT
 JSR R1, 00WRDATA
 .WORD 0
 .WORD 0
 BR OUT
 JSR R1, 00REDATA
 .WORD 0
 .WORD 0
 OUT:
 NOV (SP)+,R5
 NOV (SP)+,R4
 NOV (SP)+,R3
 NOV (SP)+,R2
 NOV (SP)+,R1
 NOV (SP)+,R0
 RTS PC

;GOING ON THE DISK

;IF "ERROR 3" INSERTED BY RDHEAD
 ;SUBROUTINE THEN THE HEADER SYNC,
 ;GENERATED BY DCL IS WRONG
 ;OR THE LAST BYTE
 ;OF THE HEADER GAP 0'S IS WRONG
 ;IN EITHER CASE WORD NO=6
 ;RIGHT BYTE IS HEADER 0
 ;LEFT BYTE IS SYNC
 ;"BAD DATA" HAS WHAT IS GOING
 ;ON DISK

;ARE ANY ERRORS DETECTED
 ;IF YES BRANCH

;IS THIS FORCED HEADER ERROR COMMAND
 ;IF YES NOSYNC=-1 THEN WRITE OR READ
 ;IS SHUT OFF SO BRANCH OUT
 ;IF NOSYNC=0 THEN CONTINUE
 ;BRANCH IF SET

;WRITE DATA
 ;NO OF WORDS TO BE WRITTEN
 ;
 ;
 ;READ DATA
 ;NO OF WORDS TO BE READ

;POP STACK INTO R5
 ;POP STACK INTO R4
 ;POP STACK INTO R3
 ;POP STACK INTO R2
 ;POP STACK INTO R1
 ;POP STACK INTO R0

;THE DISK SECTOR IS DIVIDED AS FOLLOWS
 ;19 WORDS OF 0, ONE WORD 144000
 ;THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC, BYTE

RSYNC: 14400
 RCYL: 0
 RSETR: 0
 RKEY1: 0
 RKEY2: 0

;5 WORDS OF 0 ONE WORD 144000
 ;THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC, BYTE
 ;THESE ARE DCL GENERATED

10095
 10096
 10097
 10098
 10099
 10100
 10101
 10102
 10103
 10104
 10105
 10106
 10107
 10108
 10109
 10110
 10111
 10112
 10113
 10114
 10115
 10116
 10117
 10118
 10119
 10120
 10121
 10122
 10123
 10124
 10125
 10126
 10127
 10128
 10129
 10130
 10131
 10132
 10133
 10134
 10135
 10136
 10137
 10138
 10139
 10140
 10141
 10142
 10143
 10144
 10145
 10146
 10147
 10148

;THERE ARE 256 WORDS OF DATA
 ;THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
 ;15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP
 ;*****

;*****
 ;READ DISK HEADER
 ;*****

031132 000000
 031134 000000
 031136 000000

NOBYNC: 0 ;FORCED HEADER ERROR = -1
 TY: 0 ;NORMAL = 0
 ERWORD: 0 ;ERROR TYPE NO,
 ;ERROR WORD NO,

031140 012137 031122
 031144 012137 031124
 031150 012137 031126
 031154 012137 031130
 031160 012137 031730
 031164 010146
 031166 013700 001652
 031172 012705 000002
 031176 012710 000001
 031202 052710 000010
 031206 052710 000002
 031212 042710 000012
 031216 000404
 031220 012710 000013
 031224 042710 000012
 031230 012702 000007
 031234 052710 000002
 031240 042710 000002
 031244 005302
 031246 001372
 031250 005305
 031252 001362
 031254 012702 000022
 031260 005037 031726
 031264 004737 031732

RDHEAD: MOV (R1)+, 00RCYL ;STORE CYLINDER ADDRESS
 MOV (R1)+, 00RSETR ;STORE SECTOR AND TRACK ADDRESS
 MOV (R1)+, 00RKEY1 ;STORE KEY1
 MOV (R1)+, 00RKEY2 ;STORE KEY2
 MOV (R1)+, 00CONPA ;STORE COMPARE OR NOT
 MOV R1,-(SP) ;PUSH R1 ON STACK
 MOV 00RHNR, R0 ;R0 CONTAINS MAINTANENCE REG.
 MOV 02, R5 ;R5 IS A COUNTER FOR WORDS
 MOV 0DMD, 0R0 ;DIAG, MODE
 BIS 0MSTCK,0R0 ;SET SECTOR FOR FIRST WORD
 BIS 0MCLK,0R0 ;SET CLOCK FOR FIRST WORD
 BIC 0MSTCKIMCLK,0R0 ;RESET SECTOR AND CLOCK
 BR 20 ;BRANCH OVER GIVING SECTOR FOR FIRST TIME
 101 MOV 0MSTCKIMCLKIDMD,0R0 ;SET SECTOR, CLOCK, DIAG, MODE, RESET INDEX
 BIC 0MSTCKIMCLK,0R0 ;RESET SECTOR, CLOCK
 201 MOV 07, R2 ;R2 IS A COUNTER FOR BYTES
 301 BIS 0MCLK, 0R0 ;SET CLOCK
 BIC 0MCLK, 0R0 ;RESET CLOCK
 DEC R2 ;BYTE COUNTER
 BNE 30 ;BRANCH IF BYTE NOT COMPLETE
 DEC R5 ;WORD COUNTER
 BNE 10 ;BRANCH IF WORD NOT COMPLETE
 MOV 010,, R2 ;NO OF WORDS OF ZEROS
 401 CLR 00WORD ;READ 0
 JSR PC, 00READ ;GO TO READ

10149	031270	005302				DEC	R2		;COUNT
10150	031272	001372				BNE	40		
10151	031274	013737	031120	031726		MOV	00RSYNC,00WORD		;SYNC, WORD
10152	031302	004737	031732			JSR	PC, 00READ		
10153	031306	032710	001000			BIT	0DTSY, 0R0		;SYNC, BYTE DETECTED?
10154	031312	001012				BNE	50		;BRANCH IF SYNC DETECTED
10155	031314	012737	000001	031136		MOV	01, 00ERWORD		;ERROR WORD NO
10156	031322	013737	031120	001124		MOV	00RSYNC,00SGDDAT		;SYNC WORD
10157	031330	012737	104002	031030		MOV	0104002,00SSYN		;INSERT "ERROR 2" IN SSYN
10158	031336	000571				BR	130		;BRANCH OUT
10159	031340	013737	031122	031726	50:	MOV	00RCYL, 00WORD		;SETUP CYLINDER
10160	031346	004737	031732			JSR	PC, 00READ		;READ
10161	031352	013737	031124	031726		MOV	00RSETR,00WORD		;SETUP SECTOR/TRACK
10162	031360	004737	031732			JSR	PC,00READ		;READ
10163	031364	013737	031126	031726		MOV	00RKEY1,00WORD		;SETUP KEY1
10164	031372	004737	031732			JSR	PC,00READ		;READ
10165	031376	013737	031130	031726		MOV	00RKEY2,00WORD		;SETUP KEY2
10166	031404	004737	031732			JSR	PC,00READ		;READ
10167	031410	013737	032716	031726		MOV	00WCRC,00WORD		;SETUP CRC
10168	031416	004737	031732			JSR	PC,00READ		;READ
10169	031422	005737	002014			TST	00TESDTE		;IS THIS A DRIVE TIMING ERROR
10170	031426	001135				BNE	130		;BRANCH OUT IF YES
10171	031430	005737	031730			TST	00COMPA		;IS THIS A READ OR WRITE COMMAND
10172	031434	001472				BEQ	110		
10173	031436	012705	032720			MOV	0HEGAP, R5		;POINTER FOR HEADER GAP
10174	031442	012702	000005			MOV	05, R2		;NO OF WORDS OF ZEROS
10175	031446	012737	000006	031136	60:	MOV	06,00ERWORD		;ERROR WORD NO SET
10176	031454	004737	032164			JSR	PC,00WRITE		;FOR HEADER GAP
10177	031460	005737	032162			TST	00WORD		;TEST WRITTEN WORD
10178	031464	001413				BEQ	70		;BRANCH IF GOOD THAT IS 0
10179	031466	160237	031136			SUB	R2,00ERWORD		;WORD NO IN ERROR
10180	031472	005037	001124			CLR	00SGDDAT		;GOOD WORD SHOULD BE 0
10181	031476	013767	032162	147422		MOV	00WORD,	00DDAT	;BAD DATA
10182	031504	012737	104003	031032		MOV	0104003,00HEDGAP		; "ERROR 2" GOES IN HEDGAP
10183	031512	000503				BR	130		;BRANCH OUT
10184	031514	013725	032162		70:	MOV	00WORD,(R5)+		;SAVE HEADER GAP
10185	031520	005302				DEC	R2		
10186	031522	001351				BNE	60		
10187	031524	004737	032164			JSR	PC, 00WRITE		;WRITE HEADER (DATA) GAP SYNC
10188	031530	023737	031120	032162		CHP	00RSYNC,00WORD		
10189	031536	001426				BEQ	100		
10190	031540	005737	031132			TST	00NOSYNC		;IS THIS FORCED HEADER ERROR COMMAND
10191									;IF YES NOSYNC=1 THEN WRITE OR READ
10192									;IS SHUT OFF SO BRANCH OUT
10193									;IF NO NOSYNC=0 THEN CONTINUE
10194	031544	001406				BEQ	140		;BRANCH IF TRUE ERROR
10195	031546	005737	032162			TST	00WORD		
10196	031552	001420				BEQ	100		;BRANCH IF GOOD
10197	031554	005037	001124			CLR	00SGDDAT		;IT SHOULD BE ZERO
10198	031560	000403				BR	150		;BRANCH TO TYPE ERROR
10199	031562	013737	031120	001124	140:	MOV	00RSYNC,00SGDDAT		;GOOD DATA
10200	031570	013737	032162	001126	150:	MOV	00WORD,00SDDAT		;BAD DATA
10201	031576	012737	000006	031136		MOV	06, 00ERWORD		
10202	031604	012737	104003	031034		MOV	0104003,00HEDSYN		

10203	031612	000443			BR	138		;BRANCH OUT
10204	031614	013725	032162		1001	MOV	00WORD,(R5)+	;SAVE DATA SYNC.
10205	031620	000440			BR	138		
10206							;READ COMMAND START FROM HERE	
10207	031622	012702	000005		1101	MOV	05, R2	
10208	031626	005067	000074		1201	CLR	WORD	
10209	031632	004767	000074			JSR	PC, READ	;READ HEADER GAP
10210	031636	005302				DEC	R2	;IS 5 HEADER GAP ZEROS COMPLETE
10211	031640	001372				BNE	128	;IF NOT BRANCH
10212	031642	013737	031120	031726		MOV	00RSYNC,00WORD	;SYNC WORD
10213	031650	004767	000056			JSR	PC, READ	;READ HEADER (DATA) SYNC)
10214	031654	005737	031132			TST	00NOSYNC	
10215	031660	001404				BEQ	168	;IF NOT ERROR COMMAND BRANCH
10216	031662	032710	001000			BIT	0DTSY,0R0	;SYNC, DETECTED
10217	031666	001415				BEQ	138	;IF ZERO BRANCH OUT
10218	031670	000403				BR	178	;IF NOT ZERO BRANCH TO ERROR
10219	031672	032710	001000		1601	BIT	0DTSY, 0R0	;SYNC, DETECTED?
10220	031676	001011				BNE	138	;BRANCH IF YES
10221	031700	012737	000006	031136	1701	MOV	06,00ERWORD	;ERROR WORD NO,
10222	031706	013737	031120	001124		MOV	00RSYNC,00SGDDAT	;SYNC WORD
10223	031714	012737	104002	031034		MOV	0104002,00HEDSYN	
10224	031722				1301			
(2)	031722	012601				MOV	(SP)+,R1	;POP STACK INTO R1
10225	031724	000201				RTS	R1	
10226								
10227								
10228								
10229								
10230								
10231								
10232								
10233								
10234								
10235								
10236								
10237								
10238								
10239								
10240								
10241	031726	000000				WORD:	0	
10242	031730	000000				COMPA:	0	
10243								
10244								
10245								
10246								
10247	031732					READ:		
(2)	031732	010246				MOV	R2,-(SP)	;PUSH R2 ON STACK
10248	031734	012705	000002			MOV	02, R5	;WORD COUNTER
10249	031740	012710	000001			MOV	0DMD, 0R0	;SET DIAG. MODE
10250	031744	006037	031726			ROR	00WORD	;CHECKING IF THERE IS A ONE
10251	031750	103002				BCC	10	;IF NO ONE BRANCH
10252	031752	052710	000020			BIS	0HRD, 0R0	;SET BIT 4 IF DATA HAS ONE
10253	031756	012702	000007		101	MOV	07, R2	;BYTE COUNTER
10254	031762	052710	000012			BIS	0NSTCK;MCLK,	0R0 ;SET CLOCK,DATA IF ANY, SECTOR

10255	031766	005737	027132		TST	00TSECCG		;IS THIS BIT TO GENERATE AND TEST ECC
10256	031772	001411			BEQ	68		;BRANCH IF NO
10257	031774	032710	000020		BIT	0MRD,0R0		;IS DATA BIT A ONE
10258	032000	001404			BEQ	58		;BRANCH IF DATA BIT IS 0
10259	032002	012737	177777	027124	MOV	0-1,00ECDATA		;ECC DATA BIT IS A ONE
10260	032010	000402			BR	68		;BRANCH
10261	032012	005037	027124	581	CLR	00ECDATA		;ECC DATA BIT IS A 0
10262	032016	012746	000001	681	MOV	0DMD, -(SP)		;KEEP ONLY DIAG, MODE
10263	032022	006037	031726		ROR	00WORD		;CHECKING IF THERE IS A ONE
10264	032026	103002			BCC	28		;IF NO ONE BRANCH
10265	032030	012716	000021		MOV	0MRD;DMD,		(SP) ;KEEP DATA AND DIAG, MODE
10266	032034	012610		281	MOV	(SP)+, 0R0		;PUT IN DATA,RESET CLOCK, SECTOR
10267	032036	005737	027132		TST	00TSECCG		;IS ECC TO BE GENERATED FOR THIS BIT
10268	032042	001404			BEQ	38		;BRANCH IF NO
10269	032044	005237	027144		INC	00DATENV		;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10270	032050	004737	027162		JSR	PC,00ECTEST		;GO TO GENERATE AND TEST ECC
10271	032054	052710	000002	381	BIS	0MCLK, 0R0		;SET CLOCK
10272	032060	005737	027132		TST	00TSECCG		;IS THIS BIT TO GENERATE ECC
10273	032064	001411			BEQ	88		;BRANCH IF NO
10274	032066	032710	000020		BIT	0MRD,0R0		;IS DATA BIT A ONE
10275	032072	001404			BEQ	78		;BRANCH IF DATA BIT IS = 0
10276	032074	012737	177777	027124	MOV	0-1,00ECDATA		;ECC DATA BIT IS A ONE
10277	032102	000402			BR	88		;BRANCH
10278	032104	005037	027124	781	CLR	00ECDATA		;ECC DATA BIT IS = 0
10279	032110	012746	000001	881	MOV	0DMD, -(SP)		;KEEP DIAG, MODE
10280	032114	006037	031726		ROR	00WORD		;CHECKING IF THERE IS A ONE
10281	032120	103002			BCC	48		;BRANCH IF NO ONE
10282	032122	012716	000021		MOV	0MRD;DMD,(SP)		;KEEP DIAG, MODE AND DATA
10283	032126	012610		481	MOV	(SP)+, 0R0		;SET DATA, DIAG, MODE, CLEAR CLOCK
10284	032130	005737	027132		TST	00TSECCG		;IS THIS BIT TO GENERATE ECC
10285	032134	001404			BEQ	98		;BRANCH IF NO
10286	032136	005237	027144		INC	00DATENV		;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10287	032142	004737	027162		JSR	PC,00ECTEST		;GO TO GENERATE AND TEST ECC
10288	032146	005302		981	DEC	R2		;BYTE COUNTER
10289	032150	001341			BNE	38		;BRANCH IF ONE BYTE NOT COMPLETE
10290	032152	005305			DEC	R5		;WORD COUNTER
10291	032154	001300			BNE	18		;BRANCH IF ONE WORD NOT COMPLETE
10292	032156	012602			MOV	(SP)+,R2		;POP STACK INTO R2
10293	032160	000207			RTS	PC		

10294
 10295
 10296
 10297
 10298
 10299
 10300
 10301
 10302
 10303
 10304
 10305
 10306
 10307
 10308

;.....
 ;WRITE ONE WORD WHICH COMES BACK IN "WORD"
 ;.....

10309	032162	000000		WWORD:	0	
10310						
10311						
10312						
10313						
10314	032164			WRITE:		
(2)	032164	010046		MOV	R0,-(SP)	;;PUSH R0 ON STACK
(2)	032166	010246		MOV	R2,-(SP)	;;PUSH R2 ON STACK
(2)	032170	010346		MOV	R3,-(SP)	;;PUSH R3 ON STACK
(2)	032172	010546		MOV	R5,-(SP)	;;PUSH R5 ON STACK
10315	032174	012705	000002	MOV	02, R5	;WORD COUNTER
10316	032200	012710	000001	MOV	01, 0R0	;SET DIAG. MODE
10317	032204	012702	000007	101 MOV	07, R2	;BYTE COUNTER
10318	032210	012710	000013	MOV	0MSTCK NCLK DMD, 0R0	;SET SECTOR AND CLOCK
10319	032214	032710	000040	BIT	0MWR, 0R0	;CHECK WRITEBIT IN MAINT. REG.
10320	032220	001406		BEQ	20	;BRANCH IF ZERO
10321	032222	012737	177777 027124	MOV	0-1,00ECDATA	;ECC DATA BIT IS A ONE
10322	032230	000261		SEC		;SET CARRY
10323	032232	006003		ROR	R3	;MOVE 1 FORWARD
10324	032234	000404		BR	30	
10325	032236	005037	027124	201 CLR	00ECDATA	;ECC DATA BIT IS = 0
10326	032242	000241		CLC		;CLEAR CARRY
10327	032244	006003		ROR	R3	;MOVE 0 FOR WWORD
10328	032246	012710	000001	301 MOV	0DMD, 0R0	;CLEAR SECTOR AND CLOCK
10329	032252	005737	027132	TST	00TSECCG	;IS THIS BIT TO GENERATE ECC
10330	032256	001404		BEQ	40	;BRANCH IF NO
10331	032260	005237	027144	INC	00DATENV	;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10332	032264	004737	027162	JSR	PC,00ECTEST	;GO TO GENERATE AND TEST ECC
10333	032270	052710	000002	401 BIS	0NCLK, 0R0	;SET CLOCK
10334	032274	032710	000040	BIT	0MWR, 0R0	;CHECK WRITE BIT IN MAINT. REG.
10335	032300	001406		BEQ	50	;BRANCH IF ZERO
10336	032302	012737	177777 027124	MOV	0-1,00ECDATA	;ECC DATA BIT IS A ONE
10337	032310	000261		SEC		;SET CARRY
10338	032312	006003		ROR	R3	;MOVE 1 FOR WWORD
10339	032314	000404		BR	60	
10340	032316	005037	027124	501 CLR	00ECDATA	;ECC DATA BIT IS ZERO
10341	032322	000241		CLC		;CLEAR CARRY
10342	032324	006003		ROR	R3	;MOVE 0 FOR WWORD
10343	032326	012710	000001	601 MOV	0DMD, 0R0	;CLEAR CLOCK
10344	032332	005737	027132	TST	00TSECCG	;IS THIS BIT TO GENERATE ECC
10345	032336	001404		BEQ	70	;BRANCH IF NO
10346	032340	005237	027144	INC	00DATENV	;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10347	032344	004737	027162	JSR	PC,00ECTEST	;GO TO GENERATE AND TEST ECC
10348	032350	005302		701 DEC	R2	;COUNT FOR BYTE END
10349	032352	001346		BNE	40	;IF NOT BYTE END BRANCH
10350	032354	005305		DEC	R5	;COUNT FOR WORD END
10351	032356	001312		BNE	10	;IF NOT WORD END BRANCH
10352	032360	010337	032162	MOV	R3, 00WWORD	;STORE WORD
10353	032364	012605		MOV	(SP)+,R5	;;POP STACK INTO R5
(2)	032366	012603		MOV	(SP)+,R3	;;POP STACK INTO R3
(2)	032370	012602		MOV	(SP)+,R2	;;POP STACK INTO R2
(2)	032372	012600		MOV	(SP)+,R0	;;POP STACK INTO R0
10354	032374	000207		RTS	PC	
10355						


```

10356
10357
10358
10359
10360
10361
10362
10363
10364
10365
10366
10367
10368
10369
10370
10371 032376 000000
10372 032400 000400
10373 032402 000000
10374 032404
10375 032404 011137 032376
10376 032410 012102
10377 032412 012137 031730
10378 032416 010046
   (2) 032420 010146
   (2) 032422 010246
   (2) 032424 010346
   (2) 032426 010446
10379 032430 012701 000016
10380 032434 012703 033742
10381 032440 012723 177777
10382 032444 005301
10383 032446 001374
10384 032450 013700 001652
10385 032454 013746 032400
10386 032460 163716 032376
10387 032464 011637 032402
10388 032470 012604
10389 032472 005737 002012
10390 032476 001403
10391 032500 012737 177777 027132
10392 032506 012703 032734
10393 032512 004737 032164
10394 032516 013723 032162
10395 032522 005302
10396 032524 001372
10397 032526 005704
10398 032530 001406
10399 032532 004737 032164
10400 032536 013723 032162
10401 032542 005304
10402 032544 001372
10403 032546 005037 027132
10404 032552 012701 000002
10405 032556 004767 177402
  
```

```

;.....
;WRITE DATA
;.....
  
```

```

COUNT: 0
FORMAT: 256.
ZWORDS: 0
WRDATA:
MOV (R1), 00COUNT ;STORE NO. OF WORDS TO BE WRITTEN
MOV (R1)+, R2 ;SAME IN R2
MOV (R1)+, 00COMPA ;COMPARE OR NOT
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 014., R1 ;NO. OF TOLERANCE GAP WORDS
MOV 0TOLGAP,R3 ;START OF TOLERANCE GAP TABLE
MOV 0-1, (R3)+ ;MAKE IT 177777
DEC R1 ;IS 14 COMPLETED
BNE 18 ;IF NO BRANCH
MOV 00RHMR, R0 ;R0 CONTAINS MAINTANENCE REG.
MOV 00FORMAT, -(SP)
SUB 00COUNT, (SP)
MOV (SP),00ZWORDS ;NO. OF ZERO WORDS TO BE WRITTEN
MOV (SP)+, R4
TST 00TSECC ;IS THIS AN ECC TEST
BEQ 70 ;BRANCH IF NO
MOV 0-1,00TSECCG ;THESE BITS ARE TO GENERATE ECC
MOV 0DISK, R3 ;SIMULATED DISK AREA
JSR PC, 00WRITE ;WRITE ON SIMULATED DISK
MOV 00WORD,(R3)+ ;STORE ON SIMULATED DISK
DEC R2
BNE 20
TST R4 ;ANY ZEROS TO BE WRITTEN
BEQ 40 ;BRANCH IF NONE TO BE WRITTEN
JSR PC, 00WRITE ;WRITE ZEROS ON SIMULATED DISK
MOV 00WORD, (R3)+ ;STORE
DEC R4
BNE 30
CLR 00TSECCG ;NO MORE ECC TO BE GENERATED
MOV 02, R1
JSR PC, WRITE ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
  
```

```

10406 032562 013723 032162            MOV    @=WORD,(R3)+    ;STORE ON WEEC1 AND WEEC2
10407 032566 005301            DEC    R1
10408 032570 001372            BNE    58
10409 032572 004767 177366            JSR    PC,        WRITE    ;WRITE DATA GAP
10410 032576 013723 032162            MOV    @=WORD,(R3)+    ;STORE
10411 032602 012701 000016            MOV    @14,,    R1
10412 032606 004737 032164            JSR    PC,        @=WRITE ;WRITE TOLERANCE GAP ZEROS
10413 032612 013723 032162            MOV    @=WORD,(R3)+    ;STORE
10414 032616 005301            DEC    R1
10415 032620 001372            BNE    68
10416 032622 012604            MOV    (SP)+,R4        ;;POP STACK INTO R4
          (2) 032624 012603            MOV    (SP)+,R3        ;;POP STACK INTO R3
          (2) 032626 012602            MOV    (SP)+,R2        ;;POP STACK INTO R2
          (2) 032630 012601            MOV    (SP)+,R1        ;;POP STACK INTO R1
          (2) 032632 012600            MOV    (SP)+,R0        ;;POP STACK INTO R0
10417 032634 000201            RTS    R1
  
```

```

;*****
;WRITE HEADER AND DATA
;
;THIS IS THE SIMULATED DISK
;ONLY ONE SECTOR OF SPACE IS ALLOWED
;*****
  
```

```

10440 032636 000023            SECGAP: .BLKW    19,            ;SECTOR GAP 38 BYTES OF 0
10441 032704 000001            WSSYNC: .BLKW    1            ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
10442 032706 000004            HEADER: .BLKW    4            ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
10443 032716 000001            WCRC:    .BLKW    1            ;CRC
10444 032720 000005            HEGAP:    .BLKW    5            ;HEADER GAP 10 BYTES OF 0
10445 032732 000001            HDMSYN: .BLKW    1            ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
10446 032734                    SILOTB:                    ;USED IN SILO TEST AS SILO TABLE
10447 032734 000400            DISK:    .BLKW    256,        ;DATA SPACE
10448 033734 000001            WECC1:    .BLKW    1            ;ECC1
10449 033736 000001            WECC2:    .BLKW    1            ;ECC2
10450 033740 000001            DTAGAP: .BLKW    1            ;DATA GAP 2 BYTES OF 0
10451 033742 000016            TOLGAP: .BLKW    14,        ;TOLERANCE GAP 28 BYTES OF 0
10452
10453
10454
10455
  
```

10456
 10457
 10458
 10459
 10460
 10461
 10462
 10463
 10464
 10465
 10466
 10467
 (2)
 (2)
 (2)
 (2)
 (2)
 10468
 10469
 10470
 10471
 10472
 10473
 10474
 10475
 10476
 10477
 10478
 10479
 10480
 10481
 10482
 10483
 10484
 10485
 10486
 10487
 10488
 10489
 10490
 10491
 10492
 10493
 10494
 10495
 10496
 10497
 10498
 10499
 10500
 10501
 10502
 10503
 10504

.....
 ;WRITE HEADER AND DATA

033776 011637 002002
 034002 162737 000004 002002
 034010 010046
 034012 010146
 034014 010246
 034016 010346
 034020 010446
 034022 010546
 034024 012777 000001 145620
 034032 052777 000001 145572
 034040 013746 034124
 034044 042716 177740
 034050 012637 034060
 034054 004137 035032
 034062 012701 000240
 034066 010137 034134
 034072 010137 034136
 034076 010137 034140
 034102 010137 034142
 034106 010137 034144
 034112 010137 034146
 034116 004137 034210
 034122 000000
 034124 000000
 034126 000000
 034130 000000
 034132 000000
 034134 000240

```

COMWHD: MOV      (SP),00PCJSR      ;SAVE PC OF JSR + 4
        SUB      04,00PCJSR      ;SAVE PC OF JSR
        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      0DMD,0RHMR      ;SET DIAGNOSTIC MODE
        BIS      0GO,0RHCS1      ;GO
        MOV      00WSECTR,      -(SP) ;GET DESIRED SECTOR/TRACK
        BIC      0177740,      (SP) ;MAKE ONLY SECTOR
        MOV      (SP)+, 00WTRK   ;SAVE SECTOR
        JSR      R1,00SEARCH     ;DO SEARCH SECTOR
        WTRK:  .WORD      0       ;SECTOR NO.
        MOV      0+NOP,R1        ;GOING TO MOVE NOPS
        MOV      R1,00SEGPER     ;NOP INTO SEGAP
        MOV      R1,00FSYNER     ;NOP INTO FSYNER
        MOV      R1,00ERHEAD     ;NOP INTO ERHEAD
        MOV      R1,00ERCRC      ;NOP INTO ERCRC
        MOV      R1,00ERHDGP     ;NOP INTO ERHDGAP
        MOV      R1,00HDESYN     ;NOP INTO HDESYN
        JSR      R1,00WRHEAD

WCYL:   0 ;CYLINDER
WSECTR: 0 ;SECTOR AND TRACK
WKEY1:  0 ;KEY1
WKEY2:  0 ;KEY2
GCRC:   0 ;GOOD CRC

SEGPER: NOP ;IF "ERROR 6" INSERTED BY
           ;WRHEAD SUBROUTINE THEN
           ;SECTOR GAP GOING ON DISK
           ;IS NOT RIGHT
           ;WORD NO, CONTAINS WHICH
           ;WORD IS WRONG THAT IS
           ;FIRST OF TENTH OR WHAT EVER NO.
           ;BAD WORD IS WHAT IS GOING ON DISK
           ;IF "ERROR 6" INSERTED BY
           ;WRHEAD SUBROUTINE THEN
           ;THE LAST 0 BYTE OF SECTOR
           ;GAP OF FIRST SYNC, BYTE
           ;AFTER SECTOR GAP IS IN
           ;ERROR
           ;WORD NO, CONTAINS 20
           ;RIGHT BYTE IS SECTOR GAP

FSYNER: NOP
  
```



```

(2) 034172 012603      NOV      (SP)+,R3      ;;POP STACK INTO R3
(2) 034174 012602      NOV      (SP)+,R2      ;;POP STACK INTO R2
(2) 034176 012601      NOV      (SP)+,R1      ;;POP STACK INTO R1
(2) 034200 012600      NOV      (SP)+,R0      ;;POP STACK INTO R0
10557 034202 000207      RTS      PC
  
```

10558
10559
10560
10561
10562
10563
10564
10565
10566
10567
10568
10569
10570
10571
10572
10573
10574
10575

```

;.....
;WRITE HEADER
;.....
  
```

;R0 = MAINT.REG.; R1 = SIMULATED DISK; R2 = BYTE COUNT; R3 = WRITE WORD; R5 = WORD COUNT

10576 034204 000000
10577 034206 000000
10578 034210 000000
10579 034212 000000
10580 034214 000000

```

SCYL: 0 ;
SSECTR: 0 ;
SKEY1: 0 ;
SKEY2: 0 ;
SCRC: 0 ;
  
```

10581
10582
10583 034216 012137 034204
10584 034222 012137 034206
10585 034226 012137 034210
10586 034232 012137 034212
10587 034236 012137 034214
10588 034242 010146
10589 034244 012701 032636
10590 034250 013700 001652
10591 034254 012710 000001
10592 034260 012705 000002
10593 034264 052710 000010
10594 034270 012710 000013
10595 034274 032710 000040
10596 034300 001403
10597 034302 000261
10598 034304 006003
10599 034306 000402
10600 034310 000241
10601 034312 006003
10602 034314 012710 000001
10603 034320 012702 000007
10604 034324 052710 000002
10605 034330 032710 000040
10606 034334 001403

```

WRHEAD: NOV      (R1)+, 00SCYL
        NOV      (R1)+, 00SSECTR
        NOV      (R1)+, 00SKEY1
        NOV      (R1)+, 00SKEY2
        NOV      (R1)+, 00SCRC
        NOV      R1,-(SP)      ;;PUSH R1 ON STACK
        NOV      @SECGAP,R1    ;;SIMULATED DISK INDICATOR
        NOV      @RHMR,R0      ;;R0 NOW HAS MAINT. REG. ADDR.
        NOV      @DND, R0      ;;SET DIAG. MODE
        NOV      @2, R5        ;;WORD COUNTER
        BIS      @MSTCK,R0     ;;SET SECTOR FOR FIRST BYTE
10:     NOV      @MSTCK|@MCLK|@DND,@R0,SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
        BIT      @MWR, R0     ;;CHECK WRITE BIT IN MAINT. REG.
        BEQ     20
        SEC
        ROR     R3
        BR      30
20:     CLC
        ROR     R3
30:     NOV      @DND, R0     ;;SET CARRY
        NOV      @7, R2      ;;MOVE ONE FORWARD
40:     BIS      @MCLK, R0    ;;
        BIT      @MWR, R0    ;;CLEAR CARRY
        BEQ     50          ;;MOVE ZERO FORWARD
                    ;;CLEAR CLOCK, SECTOR
                    ;;BYTE COUNTER
                    ;;SET CLOCK
                    ;;CHECK WRITE BIT IN MAINT.REG.
                    ;;BRANCH IF ZERO
  
```


10661	034606	160237	031136			SUB	R2,00ERWORD	;WORD NO
10662	034612	014337	001124			MOV	-(R3),00SGDDAT	;GOOD DATA
10663	034616	014137	001126			MOV	-(R1),008BDDAT	;BAD DATA
10664	034622	012737	104006	034140		MOV	0104006,00ERHEAD	;INSERT "ERROR 6"
10665	034630	000475				BR	170	;BRANCH OUT
10666	034632	005302			1401	DEC	R2	;ARE 4 HEADER WORDS DONE?
10667	034634	001353				BNE	130	;IF NOT BRANCH
10668	034636	004737	032164			JSR	PC, 00WRITE	;WRITE CRC
10669	034642	013711	032162			MOV	00HWORD,(R1)	;STORE CRC
10670	034646	022137	034132			CMP	(R1)+,00GCRC	;COMPARE GOOD CRC
10671	034652	001414				BEQ	200	;BRANCH IF GOOD
10672	034654	014137	001126			MOV	-(R1),008BDDATA	;BAD CRC WRITTEN
10673	034660	013737	034132	001124		MOV	00GCRC,00SGDDAT	;GOOD CRC
10674	034666	012737	000005	031136		MOV	05,00ERWORD	;ERROR WORD NO
10675	034674	012737	104006	034142		MOV	0104006,00ERCRC	;INSERT ERROR 6
10676	034702	000450				BR	170	
10677	034704	012702	000005		2001	MOV	05, R2	;NO OF HEADER GAP
10678	034710	012737	000006	031136	1501	MOV	06,00ERWORD	;ERROR WORD NO SET
10679	034716	004737	032164			JSR	PC,00WRITE	;WRITE HEADER GAP
10680	034722	013721	032162			MOV	00HWORD,(R1)+	;STORE
10681	034726	001412				BEQ	160	;IF GOOD BRANCH
10682	034730	160237	031136			SUB	R2,00ERWORD	;ERROR WORD NO
10683	034734	005037	001124			CLR	00SGDDAT	;GOOD DATA
10684	034740	014137	001126			MOV	-(R1),008BDDAT	;BAD DATA
10685	034744	012737	104006	034144		MOV	0104006,00ERHDP	;STORE "ERROR 6"
10686	034752	000424				BR	170	;BRANCH OUT
10687	034754	005302			1601	DEC	R2	;ARE 5 HEADER GAP ZEROS DONE
10688	034756	001354				BNE	150	;IF NOT BRANCH
10689	034760	004737	032164			JSR	PC,00WRITE	
10690	034764	013711	032162			MOV	00HWORD,(R1)	
10691	034770	023721	031120			CMP	00RSYNC,(R1)+	
10692	034774	001413				BEQ	170	
10693	034776	012737	000005	031136		MOV	05,00ERWORD	
10694	035004	014137	001126			MOV	-(R1), 008BDDAT	
10695	035010	013737	031120	001124		MOV	00RSYNC, 00SGDDAT	
10696	035016	012737	104006	034146		MOV	0104006,00HDESYN	
10697	035024				1701			
(2)	035024	012601				MOV	(SP)+,R1	;POP STACK INTO R1
10698	035026	000201				RTS	R1	

10699
 10700
 10701
 10702
 10703
 10704
 10705
 10706
 10707
 10708
 10709
 10710
 10711
 10712
 10713

;*****
 ;SEARCH SECTOR
 ;*****

10714
 10715
 10716
 10717
 10718
 10719
 10720
 10721
 10722
 10723
 10724
 10725
 10726
 10727
 10728
 10729
 10730
 10731
 10732
 10733
 10734
 10735
 10736
 10737
 10738 035030 000000
 10739
 10740 035032 012137 035030
 10741 035036 010046
 (2) 035040 010146
 (2) 035042 010246
 (2) 035044 010346
 (2) 035046 010446
 (2) 035050 010546
 10742 035052 013700 001652
 10743 035056 013703 035030
 10744 035062 012710 000001
 10745 035066 052710 000010
 10746 035072 042710 000010
 10747 035076 052710 000010
 10748 035102 042710 000010
 10749
 10750
 10751
 10752 035106 052710 000014
 10753 035112 012710 000001
 10754 035116 005703
 10755 035120 001461
 10756
 10757
 10758
 10759
 10760
 10761
 10762

```

)      R0=RHMR ADDRESS
)      R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
)      R2=CLOCK COUNT (PER BYTE)
)      R3=SECTOR COUNTER FROM R1
)      R5=BYTES PER WORD COUNT
)BEFORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
)SECTOR PULSE IN CASE IT IS SET
)AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE
)BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS
)IDENTICAL WITH CLOCK
)NUMBERING THE SECTOR CLOCKS AS FOLLOWS
)THE SECTOR CLOCK UNDER INDEX = 0
)THE NEXT = 1
)THE NEXT = 2
)ETC,
)THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER = 600
)THE NEXT SECTOR THEN HAS 600 SECTOR CLOCKS
)THE NEXT SECTOR THEN HAS ANOTHER 600 SECTOR CLOCKS
)AND SO ON
  
```

```

SECTR: 0                                ;SECTOR SEARCHED FOR
SEARCH: MOV        (R1)+, 00SECTR ;SAVE SECTOR SEARCHED FOR
         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        00RHMR, R0       ;NOW R0 HAS MAINTENANCE REG. ADR.
         MOV        00SECTR,R3       ;SECTOR COUNTER
         MOV        00DND, 000       ;SET DIAGNOSTIC MODE
         BIS        00NSTCK, 000     ;SET SECTOR CLOCK
         BIC        00NSTCK, 000     ;CLEAR SECTOR CLOCK
         BIS        00NSTCK, 000     ;SET SECTOR CLOCK
         BIC        00NSTCK, 000     ;CLEAR SECTOR CLOCK
         ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
         ;RESETTING SECTOR PULSE
         ;IN CASE IT STARTS SET
         BIS        00MINXINSTCK,000 ;SET INDEX AND SECTOR CLOCK
         MOV        00DND, 000       ;RESET INDEX AND SECTOR CLOCK
         TST        R3               ;IF SECTOR REQUIRED JUMP OUT
         BEQ        70               ;BRANCH OF SECTOR ZERO REQUIRED
         ;NOW THE 304 WORDS WILL START
  
```

```

)FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH
)BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN
)FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK
  
```



```

10763                                     ;WILL BE IDENTICAL WITH ONE CLOCK
10764
10765
10766                                     ;ONE WORD ONLY
10767
10768 035122 012702 000010          18:  MOV    00,,  R2      ;BYTE COUNTER
10769 035126 012705 000002          MOV    02,  R5      ;BYTES PER WORD
10770 035132 052710 000010          BIS    0NSTCK,0R0   ;SET SECTOR CLOCK
10771 035136 052710 000002          BIS    0MCLK,0R0   ;SET CLOCK
10772 035142 000402                BR     30           ;BRANCH TO CLEAR SECTOR AND CLOCK
10773 035144 052710 000012          20:  BIS    0NSTCK|MCLK,0R0 ;SET SECTOR AND CLOCK
10774 035150 042710 000012          30:  BIC    0NSTCK|MCLK,0R0 ;CLEAR SECTOR AND CLOCK
10775 035154 052710 000002          00:  BIS    0MCLK, 0R0   ;SET CLOCK
10776 035160 042710 000002          BIC    0MCLK, 0R0   ;CLEAR CLOCK
10777 035164 005302                DEC    R2          ;BYTE COUNTER
10778 035166 001372                BNE    00         ;BRANCH IF BYTE NOT COMPLETE
10779 035170 012702 000007          MOV    07,  R2      ;SETUP FOR SECOND BYTE
10780 035174 005305                DEC    R5          ;IS WORD COMPLETE?
10781 035176 001362                BNE    20         ;BRANCH IF NOT COMPLETE
10782                                     ;TO GIVE SECTOR CLOCK AND CLOCK
10783
10784
10785                                     ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL
10786
10787 035200 012701 000457          40:  MOV    0303,, R1   ;WORDS PER SECTOR COUNTER
10788 035204 012705 000002          50:  MOV    02,  R5   ;BYTES PER WORD COUNTER
10789 035210 012702 000007          MOV    07,  R2   ;BYTE COUNTER (CLOCK COUNTER)
10790 035214 052710 000012          BIS    0NSTCK|MCLK,0R0 ;SET SECTOR CLOCK AND CLOCK
10791 035220 042710 000012          BIC    0NSTCK|MCLK,0R0 ;CLEAR SECTOR CLOCK AND CLOCK
10792 035224 052710 000002          60:  BIS    0MCLK, 0R0   ;SET CLOCK
10793 035230 042710 000002          BIC    0MCLK, 0R0   ;RESET CLOCK
10794 035234 005302                DEC    R2          ;IS BYTE COMPLETE?
10795 035236 001372                BNE    60         ;BRANCH IF NOT COMPLETE
10796 035240 005305                DEC    R5          ;IS WORD COMPLETE?
10797 035242 001362                BNE    50         ;BRANCH IF NOT
10798 035244 005301                DEC    R1          ;IS SECTOR COMPLETE
10799 035246 001356                BNE    40         ;BRANCH IF NOT
10800 035250 052710 000010          BIS    0NSTCK,0R0   ;SET SECTOR
10801 035254 042710 000010          BIC    0NSTCK,0R0   ;CLEAR SECTOR
10802 035260 005303                DEC    R3          ;IS REQUIRED NO OF SECTORS COMPLETE
10803 035262 001317                BNE    10         ;BRANCH IF NOT
10804
10805 035264          70:
10806 (2) 035264 012605                MOV    (SP)+,R5    ;;POP STACK INTO R5
10807 (2) 035266 012604                MOV    (SP)+,R4    ;;POP STACK INTO R4
10808 (2) 035270 012603                MOV    (SP)+,R3    ;;POP STACK INTO R3
10809 (2) 035272 012602                MOV    (SP)+,R2    ;;POP STACK INTO R2
10810 (2) 035274 012601                MOV    (SP)+,R1    ;;POP STACK INTO R1
10811 (2) 035276 012600                MOV    (SP)+,R0    ;;POP STACK INTO R0
10812 035300 000201                RTS    R1
;*****
;READ ONE SECTOR OF DATA

```

```

10011 ;*****
10012
10013 035302 000000 RNO: 0 ;NO. OF WORDS READ
10014 035304 000000 RCOM: 0 ;EXTRA STORAGE
10015
10016
10017
10018 035306 012137 035302 REDATA: MOV (R1)+,00RNO ;SAVE NO. OF WORDS ONLY FOR INFORMATION
10019 035312 012137 035304 MOV (R1)+,00RCOM ;EXTRA WORD ONLY FOR INFORMATION
10020 035316 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
10021 035320 005737 002012 TST 00TSECC ;IS THIS AN ECC TEST
10022 035324 001403 BEQ 18 ;BRANCH IF NO
10023 035326 012737 177777 027132 MOV 0-1,00TSECCG ;THESE BITS ARE TO GENERATE ECC
10024 035334 012702 000402 100: MOV 0250,,R2 ;256 WORDS PER SECTOR
10025 ;PLUS 2 ECC WORDS
10026 035340 012703 032734 MOV 0DISK,R3 ;POINTE TO DISK SIMULATION
10027 035344 012337 031726 200: MOV (R3)+,00WORD ;READY TO READ CONTENTS
10028 035350 004737 031732 JSR PC,00READ ;READ
10029 035354 005302 DEC R2 ;IS 256 WORDS DONE?
10030 035356 001372 BNE 28 ;IF NOT BRANCH
10031 035360 005737 002012 TST 00TSECC ;IS THIS AN ECC TEST
10032 035364 001012 BNE 48 ;BRANCH OUT IF YES
10033 035366 005037 027132 CLR 00TSECCG ;NO MORE ECC BITS ARE TO BE GENERATED
10034 035372 012702 000017 MOV 015,,R2 ;ONE DATA GAP, 14 TOLERANCE GAP
10035 035376 012337 031726 300: MOV (R3)+,00WORD ;READY TO READ CONTENTS OF WORD
10036 035402 004737 031732 JSR PC,00READ ;READ
10037 035406 005302 DEC R2 ;COUNT
10038 035410 001372 BNE 38 ;BRANCH IF 14 NOT DONE
10039 035412 400: MOV (SP)+,R1 ;;POP STACK INTO R1
(P) 035412 012601 RTS R1 ;RETURN
10040 035414 000201
10041
10042
10043
10044
10045 035416 RPVECT: TYPE ,,+4 ;;TYPE ASCIZ STRING
(1) 035416 104400 035424 BR 648 ;;GET OVER THE ASCIZ
(1) 035422 000411 ;;,ASCIZ /TRAPED FROM PC = /
(1) 035446 640:
10046 035446 104402 TYPOC ;TYPE FROM PC
10047 035450 012777 035416 144142 MOV 0RPVECT,0RPVEC ;RESTORE TRAP RP04 VECTOR
10048 035456 000000 HALT ;CHANGE TO CONTINUE
10049
10050
10051
10052

```


(1)	035622	105067	143255	40:	CLRB	8ERFLG	;;ZERO THE ERROR FLAG
(1)	035626	005067	143352		CLR	8TIMES	;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1)	035632	000415			BR	10	;;ESCAPE TO THE NEXT TEST
(1)	035634	032737	004000	177570	30:	BIT	8BIT11,08SWR
(1)	035642	001011			BNE	10	;;INHIBIT ITERATIONS?
(1)	035644	005767	143230		TST	8PASS	;;BR IF YES
(1)	035650	001406			BEQ	10	;;IF FIRST PASS OF PROGRAM
(1)	035652	005267	143226		INC	8ICNT	;; INHIBIT ITERATIONS
(1)	035656	026767	143322	143220	CMP	8TIMES,8ICNT	;;INCREMENT ITERATION COUNT
(1)	035664	002021			BGE	8OVER	;;CHECK THE NUMBER OF ITERATIONS MADE
(1)	035666	012767	000001	143210	10:	MOV	81,8ICNT
(1)	035674	016767	000044	143302		MOV	8MXCNT,8TIMES
(1)	035702	105267	143174		88VLAD:	INCB	;;REINITIALIZE THE ITERATION COUNTER
(1)	035706	011667	143174			MOV	8STNM
(1)	035712	011667	143172			MOV	(8P),8LPADR
(1)	035716	005067	143264			MOV	(8P),8LPERR
(1)	035722	112767	000001	143165		CLR	;;SAVE SCOPE LOOP ADDRESS
(1)	035730	016737	143146	177570	8OVER:	MOV	8ESCAPE
(1)	035736	016716	143144			MOV	81,8ERMAX
(1)	035742	000002				MOV	8STNM,08DISPLAY
(1)	035744	000004				MOV	8LPADR,(8P)
						RTI	;;CLEAR THE ESCAPE FROM ERROR ADDRESS
							;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
							;;DISPLAY TEST NUMBER
							;;FIXES PS
							;;FUDGE RETURN ADDRESS
							;;MAX, NUMBER OF ITERATIONS


```

(3) 036122 012605          MOV      (SP)+,R5          ;;POP STACK INTO R5
(3) 036124 012603          MOV      (SP)+,R3          ;;POP STACK INTO R3
(3) 036126 012602          MOV      (SP)+,R2          ;;POP STACK INTO R2
(3) 036130 012601          MOV      (SP)+,R1          ;;POP STACK INTO R1
(3) 036132 012600          MOV      (SP)+,R0          ;;POP STACK INTO R0
(1) 036134 104400 036162    TYPE      ,0DBLK          ;;NON TYPE THE NUMBER
(1) 036140 016666 000002 000004  MOV      2(SP),4(SP)      ;;ADJUST THE STACK
(1) 036146 012616          MOV      (SP)+,(SP)
(1) 036150 000002          RTI
(1) 036152 023420          ;;RETURN TO USER
(1) 036154 001750          SDTBL: 10000.
(1) 036156 000144          1000.
(1) 036160 000012          100.
(1) 036162 000004          10.
10858 036162 000004          SDBLK: ,BLKW 4
(1)  ;.....
(1)  ,SBTTL TYPE ROUTINE
(1)  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)  ;*NOTE1:  SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)  ;*NOTE2:  SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)  ;*NOTE3:  SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)  ;*
(1)  ;*CALL:
(1)  ;*1) USING A TRAP INSTRUCTION
(1)  ;* TYPE ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)  ;*OR
(1)  ;* TYPE
(1)  ;* MESADR
(1)  ;*
(1)  ;*2) USING A JSR INSTRUCTION
(1)  ;* MOV PS,-(SP)          ;;PUSH PROCESSOR STATUS WORD ON THE STACK
(1)  ;* JSR PC,STYPE          ;;CALL TYPE ROUTINE
(1)  ;* MESADDR              ;;FIRST ADDRESS OF MESSAGE
(1)  036172 105767 142753  STYPE: TSTB  STPFLG          ;;IS THERE A TERMINAL?
(1)  036176 100002          BPL  10          ;;BR IF YES
(1)  036200 000000          HALT          ;;HALT HERE IF NO TERMINAL
(1)  036202 000407          BR  30          ;;LEAVE
(1)  036204 010046          10: MOV  R0,-(SP)          ;;SAVE R0
(1)  036206 017600 000002  MOV  02(SP),R0          ;;GET ADDRESS OF ASCIZ STRING
(1)  036212 112046          20: MOV  (R0)+,-(SP)          ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  036214 001005          BNE  40          ;;BR IF IT ISN'T THE TERMINATOR
(1)  036216 005726          TST  (SP)+          ;;IF TERMINATOR POP IT OFF THE STACK
(1)  036220 012600          MOV  (SP)+,R0          ;;RESTORE R0
(1)  036222 062716 000002  30: ADD  02,(SP)          ;;ADJUST RETURN PC
(1)  036226 000002          RTI          ;;RETURN
(1)  036230 122716 000011  40: CMPB 0HT,(SP)          ;;BRANCH IF <HT>
(1)  036234 001424          BEQ  00
(1)  036236 122716 000200  CMPB 0CRLF,(SP)          ;;BRANCH IF NOT
(1)  036242 001004          BNE  50
(1)  036244 005726          TST  (SP)+          ;;POP <CR><LF> EQUIV
(1)  036246 104400 001215  TYPE, 0CRLF

```



```

(1)
(1) 036424 005067 177754          ;
(1) 036430 012767 036412 177750  STKINT: CLR      STKCNT      ;;CLEAR COUNT OF ITEMS IN QUEUE
(1) 036436 016767 177744 177744  MOV      08TKQSR,STKQIN ;;MOVE THE STARTING ADDRESS OF THE
(1) 036444 012737 036474 000060  MOV      STKQIN,STKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS,
(1) 036452 012737 000200 000062  MOV      08TKSRV,08TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
(1) 036460 005777 142454          MOV      0200,08TKVEC+2 ;;"BR" LEVEL 4
(1) 036464 012777 000100 142444  TST      08TKB        ;;CLEAR DONE FLAG
(1) 036472 000207          MOV      0100,08TKS   ;;ENABLE TTY KEYBOARD INTERRUPT
(1)                                RTS      PC          ;;RETURN TO CALLER
(1)
(1)                                ;*TK SERVICE ROUTINE
(1)                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
(1)                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
(1)                                ;*IT IN THE QUEUE,
(1)                                ;*IF THE CHARACTER IS A "CONTROL-C" ("C) STKINT IS CALLED AND
(1)                                ;*UPON RETURN EXIT IS MAKE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
(1)                                ;
(1) 036474 117746 142440          STKSRV: MOVB   08TKB,-(SP)    ;;PICKUP THE CHARACTER
(1) 036500 042716 177600          BIC      0"C177,(SP)       ;;STRIP THE JUNK
(1) 036504 021627 000003          CMP      (SP),03          ;;IS IT A CONTROL C?
(1) 036510 001006          BNE      10                ;;BRANCH IF NO
(1) 036512 104400 037005          TYPE    ,0CNTLC          ;;TYPE A CONTROL-C ("C)
(1) 036516 004767 177702          JSR     PC,STKINT         ;;INIT THE KEYBOARD
(1) 036522 000167 164662          JMP     OPERSEL          ;;CONTROL C RESTART
(1) 036526 022767 000011 177650 10:  CMP      09,,STKCNT       ;;IS THE QUEUE FULL?
(1) 036534 001004          BNE      20                ;;BRANCH IF NO
(1) 036536 104400 001210          TYPE    ,0BELL           ;;RING THE TTY BELL
(1) 036542 005726          TST     (SP)+            ;;CLEAN CHARACTER OFF OF STACK
(1) 036544 000415          BR      30                ;;EXIT
(1) 036546 005267 177632          20:  INC      STKCNT        ;;COUNT THIS CHARACTER
(1) 036552 112677 177630          MOVB   (SP)+,08TKQIN     ;;AND PUT IT IN QUEUE
(1) 036556 005267 177624          INC     STKQIN           ;;UPDATE THE POINTER
(1) 036562 026727 177620 036423  CMP     STKQIN,08TKQEND  ;;GO OFF THE END?
(1) 036570 001003          BNE     30                ;;BRANCH IF NO
(1) 036572 012767 036412 177606  MOV     08TKQSR,STKQIN  ;;RESET THE POINTER
(1) 036600 000002          30:  RTI                    ;;RETURN
(1)                                ;*****
(1)                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                ;*CALL:
(1)                                ;*
(1)                                ;*   RDCHR          ;;GET A CHARACTER FROM THE QUEUE
(1)                                ;*   RETURN HERE    ;;CHARACTER IS ON THE STACK
(1)                                ;*   WITH PARITY BIT STRIPPED OFF
(1)                                ;
(1)                                ;
(1) 036602 011646          BRDCHR: MOV     (SP),-(SP)  ;;PUSH DOWN THE PC AND
(1) 036604 016666 000004 000002  MOV     4(SP),2(SP)      ;;THE PS
(1) 036612 005066 000004          CLR     4(SP)           ;;GET READY FOR A CHARACTER
(1) 036616 005037 177776          CLR     00PS            ;;ALLOW INTERRUPTS
(1) 036622 005767 177556          10:  TST     STKCNT        ;;WAIT ON A CHARACTER
(1) 036626 001775          BEQ     10                ;;
(1) 036630 005367 177550          DEC     STKCNT          ;;DECREMENT THE COUNTER
(1) 036634 117766 177550 000004  MOVB   08TKQOUT,4(SP)   ;;GET ONE CHARACTER
(1) 036642 005267 177542          INC     STKQOUT         ;;UPDATE THE POINTER
(1) 036646 026727 177536 036423  CMP     STKQOUT,08TKQEND ;;DID IT GO OFF OF THE END?

```


(1)	037012	011646		BRDOCT:	MOV	(SP),-(SP)	;;PROVIDE SPACE FOR THE
(1)	037014	016666	000004		MOV	4(SP),2(SP)	;;INPUT NUMBER
(3)	037022	010046			MOV	R0,-(SP)	;;PUSH R0 ON STACK
(3)	037024	010146			MOV	R1,-(SP)	;;PUSH R1 ON STACK
(3)	037026	010246			MOV	R2,-(SP)	;;PUSH R2 ON STACK
(1)	037030	104414		10:	RDLIN		;;READ AN ASCII LINE
(1)	037032	012600			MOV	(SP)+,R0	;;GET ADDRESS OF 1ST CHARACTER
(1)	037034	010067	000100		MOV	R0,56	;;AND SAVE IT
(1)	037040	005001			CLR	R1	;;CLEAR DATA WORD
(1)	037042	005002			CLR	R2	
(1)	037044	112046		20:	MOVB	(R0)+,-(SP)	;;PICKUP THIS CHARACTER
(1)	037046	001420			BEG	38	;;IF ZERO GET OUT
(1)	037050	122716	000060		CMPB	0'0,(SP)	;;MAKE SURE THIS CHARACTER
(1)	037054	003026			BGT	48	;;IS AN OCTAL DIGIT
(1)	037056	122716	000067		CMPB	0'7,(SP)	
(1)	037062	002423			BLT	48	
(1)	037064	006301			ASL	R1	;;*2
(1)	037066	006102			ROL	R2	
(1)	037070	006301			ASL	R1	;;*4
(1)	037072	006102			ROL	R2	
(1)	037074	006301			ASL	R1	;;*8
(1)	037076	006102			ROL	R2	
(1)	037100	042716	177770		BIC	0'C7,(SP)	;;STRIP THE ASCII JUNK
(1)	037104	062601			ADD	(SP)+,R1	;;ADD IN THIS DIGIT
(1)	037106	000756			BR	28	;;LOOP
(1)	037110	005726		30:	TST	(SP)+	;;CLEAN TERMINATOR FROM STACK
(1)	037112	010166	000012		MOV	R1,12(SP)	;;SAVE THE RESULT
(1)	037116	010267	000026		MOV	R2,8HIOCT	
(3)	037122	012602			MOV	(SP)+,R2	;;POP STACK INTO R2
(3)	037124	012601			MOV	(SP)+,R1	;;POP STACK INTO R1
(3)	037126	012600			MOV	(SP)+,R0	;;POP STACK INTO R0
(1)	037130	000002			RTI		;;RETURN
(1)	037132	005726		40:	TST	(SP)+	;;CLEAN PARTIAL FROM STACK
(1)	037134	105010			CLRB	(R0)	;;SET A TERMINATOR
(1)	037136	104400			TYPE		;;TYPE UP THRU THE BAD CHAR.
(1)	037140	000000		50:	,WORD	0	
(1)	037142	104400	001214		TYPE	,8QUES	;;"?" "CR" & "LF"
(1)	037146	000730			BR	18	;;TRY AGAIN
(1)	037150	000000		8HIOCT:	,WORD	0	;;HIGH ORDER BITS GO HERE

```

10866 ;*****
(1)
(1) ;SBTTL ERROR HANDLER ROUTINE
(1)
(1) ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1) ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1) ;*AND GO TO SERRTYP ON ERROR
(1) ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;*SW15=1 HALT ON ERROR
(1) ;*SW13=1 INHIBIT ERROR TYPEOUTS
(1) ;*SW10=1 BELL ON ERROR
(1) ;*SW09=1 LOOP ON ERROR
(1) ;*CALL
(1) ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1)
(1) 037152 SERROR:
(3)
(3) 037152 012737 177777 001774 REGSA11 MCV 0-1,0=ERFLG ;SET ERROR FLAG
(3) 037160
(3)
(1) 037160 105267 141717 78: INCB SERFLG ;;SET THE ERROR FLAG
(1) 037164 001775 BEQ 78 ;;DON'T LET THE FLAG GO TO ZERO
(1) 037166 016737 141710 177570 MOV $STNM,0=DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1) 037174 032737 002000 177570 BIT $BIT10,0=SWR ;;BELL ON ERROR?
(1) 037202 001402 BEQ 10 ;;NO - SKIP
(1) 037204 104400 001210 TYPE ,SBELL ;;RING BELL
(1) 037210 005267 141676 10: INC SERTTL ;;COUNT THE NUMBER OF ERRORS
(1) 037214 011667 141676 MOV (SP),SERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
(1) 037220 162767 000002 141670 SUB 02,SERRPC
(1) 037226 117767 141664 141660 MOVB 0=SERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
(1) 037234 032737 020000 177570 BIT $BIT13,0=SWR ;;SKIP TYPEOUT IF SET
(1) 037242 001004 BNE 20 ;;SKIP TYPEOUTS
(1) 037244 004737 037314 JSR PC,0=SERRTYP ;;GO TO USER ERROR ROUTINE
(1) 037250 104400 001215 TYPE ,SCLRF
(1) 037254 005737 177570 20: TST 0=SWR ;;HALT ON ERROR
(1) 037260 100001 BPL 30 ;;SKIP IF CONTINUE
(1) 037262 000000 HALT ;;HALT ON ERROR!
(1) 037264 032737 001000 177570 30: BIT $BIT09,0=SWR ;;LOOP ON ERROR SWITCH SET?
(1) 037272 001402 BEQ 40 ;;BR IF NO
(1) 037274 016716 141610 MOV $LPERR,(SP) ;;FUDGE RETURN FOR LOOPING
(1) 037300 005767 141702 40: TST $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
(1) 037304 001402 BEQ 50 ;;BR IF NONE
(1) 037306 016716 141674 MOV $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1) 037312 50:
(1) 037312 000002 RTI ;;RETURN
10867 ;*****
(1)
(1) ;SBTTL ERROR MESSAGE TYPEOUT ROUTINE
(1)
(1) ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1) ;*ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1) ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR,
(1)

```



```

(1) ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
(1) ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1) ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1) ;*CALL:
(1) ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) ;*      TYPOS      ;;CALL FOR TYPEOUT
(1) ;*      ,BYTE  N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1) ;*      ,BYTE  M      ;;M=1 OR 0
(1) ;*                               ;;1=TYPE LEADING ZEROS
(1) ;*                               ;;0=SUPPRESS LEADING ZEROS
(1) ;*
(1) ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1) ;*STYPOS OR STYPOC
(1) ;*CALL:
(1) ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) ;*      TYPON      ;;CALL FOR TYPEOUT
(1) ;*
(1) ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1) ;*CALL:
(1) ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) ;*      TYPOC      ;;CALL FOR TYPEOUT
(1)
(1) 037470 017646 000000      STYPOS: MOV      0(SP),-(SP)      ;;PICKUP THE MODE
(1) 037474 116667 000001 000211      MOV      1(SP),00FILL      ;;LOAD ZERO FILL SWITCH
(1) 037502 112667 000207      MOV      (SP)+,00MODE+1      ;;NUMBER OF DIGITS TO TYPE
(1) 037506 062716 000002      ADD      02,(SP)      ;;ADJUST RETURN ADDRESS
(1) 037512 000406      BR      STYPON
(1) 037514 112767 000001 000171      STYPOC: MOV      01,00FILL      ;;SET THE ZERO FILL SWITCH
(1) 037522 112767 000006 000165      MOV      06,00MODE+1      ;;SET FOR SIX(6) DIGITS
(1) 037530 112767 000005 000154      STYPON: MOV      05,00CNT      ;;SET THE ITERATION COUNT
(1) 037536 010346      MOV      R3,-(SP)      ;;SAVE R3
(1) 037540 010446      MOV      R4,-(SP)      ;;SAVE R4
(1) 037542 010546      MOV      R5,-(SP)      ;;SAVE R5
(1) 037544 116704 000145      MOV      00MODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
(1) 037550 005404      NEG      R4
(1) 037552 062704 000006      ADD      06,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
(1) 037556 110467 000132      MOV      R4,00MODE      ;;SAVE IT FOR USE
(1) 037562 116704 000125      MOV      00FILL,R4      ;;GET THE ZERO FILL SWITCH
(1) 037566 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
(1) 037572 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
(1) 037574 006105      101    ROL      R5      ;;ROTATE MSB INTO "C"
(1) 037576 000404      BR      30      ;;GO DO MSB
(1) 037600 006105      201    ROL      R5      ;;FORM THIS DIGIT
(1) 037602 006105      ROL      R5
(1) 037604 006105      ROL      R5
(1) 037606 010503      MOV      R5,R3
(1) 037610 006103      301    ROL      R3      ;;GET LSB OF THIS DIGIT
(1) 037612 105367 000076      DECB    00MODE      ;;TYPE THIS DIGIT?
(1) 037616 100016      BPL     70      ;;BR IF NO
(1) 037620 042703 177770      BIC     0177770,R3      ;;GET RID OF JUNK
(1) 037624 001002      BNE     40      ;;TEST FOR 0
(1) 037626 005704      TST     R4      ;;SUPPRESS THIS 0?
(1) 037630 001403      BEQ     50      ;;BR IF YES
(1) 037632 005204      401    INC     R4      ;;DON'T SUPPRESS ANYMORE 0'S

```

(1)	037634	052703	000060		BIS	0'0,R3	;;MAKE THIS DIGIT ASCII
(1)	037640	052703	000040	58:	BIS	0' ,R3	;;MAKE ASCII IF NOT ALREADY
(1)	037644	110367	000040		MOVB	R3,00	;;SAVE FOR TYPING
(1)	037650	104400	037710		TYPE	,00	;;GO TYPE THIS DIGIT
(1)	037654	105367	000032	78:	DECB	00CNT	;;COUNT BY 1
(1)	037660	003347			BGT	28	;;BR IF MORE TO DO
(1)	037662	002402			BLT	60	;;BR IF DONE
(1)	037664	005204			INC	R4	;;INSURE LAST DIGIT ISN'T A BLANK
(1)	037666	000744			BR	28	;;GO DO THE LAST DIGIT
(1)	037670	012605		68:	MOV	(SP)+,R5	;;RESTORE R5
(1)	037672	012604			MOV	(SP)+,R4	;;RESTORE R4
(1)	037674	012603			MOV	(SP)+,R3	;;RESTORE R3
(1)	037676	016666	000002 000004		MOV	2(SP),4(SP)	;;SET THE STACK FOR RETURNING
(1)	037704	012616			MOV	(SP)+,(SP)	
(1)	037706	000002			RTI		;;RETURN
(1)	037710	000		88:	,BYTE	0	;;STORAGE FOR ASCII DIGIT
(1)	037711	000			,BYTE	0	;;TERMINATOR FOR TYPE ROUTINE
(1)	037712	000		00CNT:	,BYTE	0	;;OCTAL DIGIT COUNTER
(1)	037713	000		00FILL:	,BYTE	0	;;ZERO FILL SWITCH
(1)	037714	000000		00MODE:	,WORD	0	;;NUMBER OF DIGITS TO TYPE

```

10871 ;*****
(1)
(1) .SBTTL TRAP DECODER
(1)
(1) ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1) ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1) ;*OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
(1) ;*GO TO THAT ROUTINE.
(1)
(1) 037716 010046          STRAP:  MOV    R0,-(SP)          ;;SAVE R0
(1) 037720 016600 000002    MOV    2(SP),R0          ;;GET TRAP ADDRESS
(1) 037724 005740          TST    -(R0)           ;;BACKUP BY 2
(1) 037726 111000          MOVB   (R0),R0         ;;GET RIGHT BYTE OF TRAP
(1) 037730 016000 037736    MOV    STRPAD(R0),R0   ;;INDEX TO TABLE
(1) 037734 000200          RTS    R0              ;;GO TO ROUTINE
(1)
(3)
(3) .SBTTL TRAP TABLE
(3)
(3) ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3) ;*BY THE "TRAP" INSTRUCTION.
(3)
(3) ;
(3) ; ROUTINE
(3) ; -----
(3) 037736          STRPAD:
(3) 037736 036172      STYPE   ;;CALL=TYPE      TRAP+0(104400) TTY TYPEOUT ROUTINE
(3) 037740 037514      STYPOC  ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3) 037742 037470      STYPOS  ;;CALL=TYPOS     TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3) 037744 037530      STYPON  ;;CALL=TYPON     TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST CALL)
(3) 037746 035746      STYPDS  ;;CALL=TYPDS     TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
(3) 037750 036602      SRDCHR  ;;CALL=RDCHR     TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE
(3) 037752 036666      SRDLIN  ;;CALL=RDLIN     TRAP+14(104414) TTY TYPEIN STRING ROUTINE
(3) 037754 037012      SRDOCT  ;;CALL=RDOCT     TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY
10872 037756 024452      T.SCOP  ;;CALL=SCOP1     TRAP+20(104420) NY LOCAL SCOPES
10873 037760 024524      CHECKT  ;;CALL=CHECKD     TRAP+22(104422) CHECK DVA,RDY,DPR,DRY
10874 037762 025026      WAIT.T  ;;CALL=WAT         TRAP+24(104424) WAIT LOOP
10875
10876
10877

```

```

10879 ;*****
(1)
(1)
(1)
(1)
(1) 037764 012737 040112 000024 SPWRDN: MOV $SILLUP,$PWRVEC ;;SET FOR FAST UP
(1) 037772 012737 000340 000026 MOV $340,$PWRVEC+2 ;;PRIO:7
(3) 040000 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
(3) 040002 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
(3) 040004 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
(3) 040006 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
(3) 040010 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
(3) 040012 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
(1) 040014 010667 000076 MOV SP,$SAVR6 ;;SAVE SP
(1) 040020 012737 040032 000024 MOV $PWRUP,$PWRVEC ;;SET UP VECTOR
(1) 040026 000000 HALT
(1) 040030 000776 BR ,-2 ;;HANG UP
(1)
(1)
(1) 040032 016706 000060 SPWRUP: MOV $SAVR6,SP ;;GET SP
(1) 040036 005067 000054 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
(1) 040042 005267 000050 10: INC $SAVR6 ;;WAIT FOR THE INC
(1) 040046 001375 BNE 10 ;;OF WORD
(3) 040050 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
(3) 040052 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
(3) 040054 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
(3) 040056 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
(3) 040060 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
(3) 040062 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
(1) 040064 012737 037764 000024 MOV $PWRDN,$PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1) 040072 012737 000340 000026 MOV $340,$PWRVEC+2 ;;PRIO:7
(1) 040100 104400 TYPE ;;REPORT THE POWER FAILURE
(1) 040102 040120 SPWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
(1) 040104 012716 MOV (PC)+,(SP) ;;RESTART AT BEGIN
(1) 040106 004220 SPWRAD: .WORD BEGIN ;;RESTART ADDRESS
(1) 040110 000002 RTI
(1) 040112 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
(1) 040114 000776 BR ,-2 ;; BEFORE THE POWER DOWN WAS COMPLETE
(1) 040116 000000 $SAVR6: 0 ;;PUT THE SP HERE
(1) 040120 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
(1) 040126 000122 .EVEN
  
```



```

10881 ;.....
10882 ;
10883 ;ERROR AND MESSAGE TABLE CONDIMENTS
10884 ;
10885 ;.....
10886
10887
10888
10889
10890 040130 051127 047117 020107 EM11 .ASCIZ /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
      040136 040504 040524 044440
      040144 020116 042522 042101
      040152 047111 020107 051117
      040160 053440 044522 044524
      040166 043516 044040 051101
      040174 053504 051101 020105
      040202 042522 044507 052123
      040210 051105 000
10891 040213 105 051122 051117 EM21 .ASCIZ /ERROR ON DATA COMMAND/
      040220 047440 020116 042040
      040226 052101 020101 047503
      040234 046515 047101 000104
10892 040242 051105 047522 020122 EM61 .ASCIZ /ERROR ON WRITE HEADER AND DATA/
      040250 047117 053440 044522
      040256 042524 044040 040505
      040264 042504 020122 047101
      040272 020104 040504 040524
      040300 000
10893 040301 103 047117 051124 EM111 .ASCIZ /CONTROLLER OR DRIVE STATUS/
      040306 046117 042514 020122
      040314 051117 042040 044522
      040322 042526 051440 040524
      040330 052524 000123
10894 040334 042522 044507 052123 EM141 .ASCIZ /REGISTER FAILED/
      040342 051105 043040 044501
      040350 042514 000104
10895 040354 050123 041505 043111 EM151 .ASCIZ /SPECIFIED REGISTER NON EXISTANT SO ABORT/
      040362 042511 020104 042522
      040370 044507 052123 051105
      040376 047040 047117 042440
      040404 044530 052123 047101
      040412 020124 047523 040440
      040420 047502 052122 000
10896 040425 127 044501 020124 EM161 .ASCIZ /WAIT LOOP FAILED/
      040432 047514 050117 043040
      040440 044501 042514 000104
10897 040446 051127 052111 020105 EM171 .ASCIZ /WRITE CHECK FAILING/
      040454 044103 041505 020113
      040462 040506 046111 047111
      040470 000107
10898 040472 042522 044507 052123 EM201 .ASCIZ /REGISTER FAILING/
      040500 051105 043040 044501
      040506 044514 043516 000
10899 040513 111 052116 051105 EM211 .ASCIZ /INTERRUPT FAILING/

```

	040520	052522	052120	043040	
	040526	044501	044514	043516	
	040534	000			
10900	040535	105	051122	051117	EM22: .ASCII /ERROR ON DRIVE PRESENT/<15><12>
	040542	047440	020116	051104	
	040550	053111	020105	051120	
	040556	051505	047105	006524	
	040564	012			
10901	040565	124	042510	052440	.ASCII /THE UNIT NO, FOUND BY SETTING RHAS/<15><12>
	040572	044516	020124	047516	
	040600	020056	047506	047125	
	040606	020104	054502	051440	
	040614	052105	044524	043516	
	040622	051040	040510	006523	
	040630	012			
10902	040631	104	020117	047516	.ASCII /DO NOT AGREE WITH THE UNIT NO, FOUND FROM/<15><12>
	040636	020124	043501	042522	
	040644	020105	044527	044124	
	040652	052040	042510	052440	
	040660	044516	020124	047516	
	040666	020056	047506	047125	
	040674	020104	051106	046517	
	040702	005015			
10903	040704	044122	051503	026462	.ASCII /RHCS2-NED BIT 012/<15><12>
	040712	042516	020104	044502	
	040720	020124	030443	006462	
	040726	012			
10904	040727	061	033467	033467	.ASCII /177777-MEANS NO UNIT FOUND/<15><12>
	040734	026467	042515	047101	
	040742	020123	047516	052440	
	040750	044516	020124	047506	
	040756	047125	006504	012	
10905	040763	116	052117	035105	.ASCII /NOTE: ON DUAL PORT SYSTEM, DRIVE ON OTHER PORT WILL NOT GIVE/<15><12>
	040770	047440	020116	052504	
	040776	046101	050040	051117	
	041004	020124	054523	052123	
	041012	046505	020054	051104	
	041020	053111	020105	047117	
	041026	047440	044124	051105	
	041034	050040	051117	020124	
	041042	044527	046114	047040	
	041050	052117	043440	053111	
	041056	006505	012		
10906	041061	116	042105	044040	.ASCII /NED HENCE THERE WILL BE AN EXTRA DRIVE/
	041066	047105	042503	052040	
	041074	042510	042522	053440	
	041102	046111	020114	042502	
	041110	040440	020116	054105	
	041116	051124	020101	051104	
	041124	053111	000105		
10907	041130	047514	045517	040440	EM24: .ASCII /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
	041136	042510	042101	051040	
	041144	043505	051511	042524	
	041152	020122	052101	052040	

	041160	042510	041040	043505	
	041166	047111	044516	043516	
	041174	047440	020106	042523	
	041202	052103	051117	044440	
	041210	020123	047111	042440	
	041216	051122	051117	000	
10908	041223	114	047517	020113	EM25: .ASCIZ /LOOK AHEAD REGISTER IS IN ERROR/
	041230	044101	040505	020104	
	041236	042522	044507	052123	
	041244	051105	044440	020123	
	041252	047111	042440	051122	
	041260	051117	000		
10909	041263	103	051125	042522	EM30: .ASCII /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGISTER/<15><12>
	041270	052116	041440	046131	
	041276	047111	042504	020122	
	041304	047504	051505	047040	
	041312	052117	046440	052101	
	041320	044103	042040	051505	
	041326	051111	042105	041440	
	041334	046131	047111	042504	
	041342	020122	042522	044507	
	041350	051510	042524	006522	
	041356	012			
10910	041357	101	052106	051105	.ASCIZ /AFTER A SEEK AND INIT/
	041364	040440	051440	042505	
	041372	020113	047101	020104	
	041400	047111	052111	000	
10911	041405	105	041503	043440	EM31: .ASCII /ECC GENERATED IS INCORRECT/<15><12>
	041412	047105	051105	052101	
	041420	042105	044440	020123	
	041426	047111	047503	051122	
	041434	041505	006524	012	
10912	041441	105	042526	054522	.ASCIZ /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
	041446	053440	051117	020104	
	041454	047117	052040	044510	
	041462	020123	042523	052103	
	041470	051117	044440	020123	
	041476	044124	052101	043440	
	041504	053111	047105	044440	
	041512	020116	042042	052101	
	041520	020101	051525	042105	
	041526	000042			
10913	041530	047117	051040	040505	EM32: .ASCII /ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ/<15><12>
	041536	020104	047503	046515	
	041544	047101	020104	043101	
	041552	042524	020122	040504	
	041560	040524	040440	042116	
	041566	042440	041503	044040	
	041574	053101	020105	042502	
	041602	047105	051040	040505	
	041610	006504	012		
10914	041613	105	041503	051040	.ASCII /ECC REGISTERS OR RHER1 IS IN ERROR/<15><12>
	041620	043505	051511	042524	
	041626	051522	047440	020122	

DERPTA.P11 POWER DOWN AND UP ROUTINES

	041634	044122	051105	020061	
	041642	051511	044440	020116	
	041650	051105	047522	006522	
	041656	012			
10915	041657	117	046116	020131	.ASCII /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
	041664	047514	042527	020122	
	041672	030461	041040	052111	
	041700	020123	043117	050040	
	041706	052101	042524	047122	
	041714	051040	043505	020056	
	041722	040503	020116	042502	
	041730	051040	040505	006504	
	041736	012			
10916	041737	124	044510	020123	.ASCIZ /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
	041744	044123	052517	042114	
	041752	046440	052101	044103	
	041760	046040	053517	051105	
	041766	030440	020061	044502	
	041774	051524	047440	020106	
	042002	047507	042117	042440	
	042010	041503	000061		
10917	042014	044510	044107	041440	EM33: .ASCIZ /HIGH COUNT BIT NOT SET AFTER 30859 CLOCKS/
	042022	052517	052116	041040	
	042030	052111	047040	052117	
	042036	051440	052105	040440	
	042044	052106	051105	031440	
	042052	034070	034468	041440	
	042060	047514	045503	000123	
10918	042066	042532	047522	042040	EM34: .ASCIZ /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
	042074	052105	041505	020124	
	042102	044502	020124	047516	
	042110	020124	044510	044107	
	042116	053440	042510	020116	
	042124	031063	041040	052111	
	042132	042440	041503	051040	
	042140	043505	020056	040510	
	042146	020123	030462	055040	
	042154	051105	051517	000	
10919	042161	120	051517	052111	EM35: .ASCII /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
	042166	047511	020116	042522	
	042174	044507	052123	051105	
	042202	047440	020122	030461	
	042210	041040	052111	020123	
	042216	043117	050040	052101	
	042224	042524	047122	051040	
	042232	043505	051511	042524	
	042240	020122	047111	047503	
	042246	051122	041505	006524	
	042254	012			
10920	042255	114	053517	051105	.ASCII /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
	042262	030440	020061	044502	
	042270	051524	047440	020106	
	042276	040520	052124	051105	
	042304	020116	042522	044507	

	042312	052123	051105	051440	
	042320	047510	046125	020104	
	042326	040515	041524	020110	
	042334	047514	042527	006522	
	042342	012			
10921	042343	061	020061	044502	.ASCII /11 BITS OF GOOD ECC1/<15><12>
	042350	051524	047440	020106	
	042356	047507	042117	042440	
	042364	041503	006461	012	
10922	042371	104	052101	042440	.ASCIZ /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
	042376	053116	047514	020120	
	042404	047507	042117	050040	
	042412	051517	052111	047511	
	042420	020116	047101	020104	
	042426	026516	047503	042504	
	042434	055040	051105	051517	
	042442	040440	042522	044440	
	042450	020116	041517	040524	
	042456	000114			
10923	042460	047117	051040	040505	EM36: .ASCII /ON READ COMMAND WITH NON-CORRECTABLE ERROR DCK AND ECH SHOULD BE SET/<1
	042466	020104	047503	046515	
	042474	047101	020104	044527	
	042502	044124	047040	047117	
	042510	041455	051117	042522	
	042516	052103	041101	042514	
	042524	042440	051122	051117	
	042532	042040	045503	040440	
	042540	042116	042440	044103	
	042546	051440	047510	046125	
	042554	020104	042502	051440	
	042562	052105	005015		
10924	042566	043111	050040	051517	.ASCIZ /IF POSITION REGISTER =10040 OR 10041 IT IS GOOD/
	042574	052111	047511	020116	
	042602	042522	044507	052123	
	042610	051105	036440	030061	
	042616	032060	020060	051117	
	042624	030440	030060	030464	
	042632	044440	020124	051511	
	042640	043440	047517	000104	
10925	042646	051127	052111	047111	EM37: .ASCIZ /WRITING WITH BUS ADDRESS HIGHER THAN 20K CAUSED ERROR/
	042654	020107	044527	044124	
	042662	041040	051525	040440	
	042670	042104	042522	051523	
	042676	044040	043511	042510	
	042704	020122	044124	047101	
	042712	031040	045470	041440	
	042720	052501	042523	020104	
	042726	051105	047522	000122	
10926	042734	020040	041520	020040	DH1: .ASCII / PC TEST REG. GOOD RECEIVED/<15><12>
	042742	020040	020040	042524	
	042750	052123	020040	020040	
	042756	042522	027107	020040	
	042764	020040	047507	042117	
	042772	020040	042522	042503	

10927	043000	053111	042105	005015															
	043006	020040	020040	020040		.ASCIZ /	NO	ADDR,	DATA	DATA /									
	043014	020040	020040	047040															
	043022	020117	020040	040440															
	043030	042104	027122	020040															
	043036	020040	040504	040524															
	043044	020040	020040	040504															
10928	043052	040524	020040	000															
	043057	040	050040	020103	DH2:	.ASCII / PC	TEST	WORD	GOOD	BAD	/ <15><12>								
	043064	020040	052040	051505															
	043072	020124	020040	020040															
	043100	053440	051117	020104															
	043106	020040	043440	047517															
	043114	020104	020040	020040															
	043122	040502	020104	006440															
	043130	012																	
10929	043131	040	020040	020040		.ASCIZ /	NO	NO	DATA	DATA /									
	043136	020040	020040	047516															
	043144	020040	020040	020040															
	043152	020040	047516	020040															
	043160	020040	042040	052101															
	043166	020101	020040	042040															
10930	043174	052101	020101	000040															
	043202	020040	041520	020040	DH11:	.ASCII / PC	TEST	FAILING	CONT.	CONT.	CONT.	CONT.	/ <15><12>						
	043210	020040	020040	042524															
	043216	052123	020040	040506															
	043224	046111	047111	020107															
	043232	041440	047117	027124															
	043240	020040	041440	047117															
	043246	027124	020040	041440															
	043254	047117	027124	020040															
	043262	041440	047117	027124															
	043270	006440	012																
10931	043273	040	020040	020040		.ASCIZ /	NO	REG	ADR	RHC81	RHC82	RHDS1	RHER1/						
	043300	020040	020040	020040															
	043306	047516	020040	051040															
	043314	043505	040440	051104															
	043322	051040	041510	030523															
	043330	020040	051040	041510															
	043336	031123	020040	051040															
	043344	042110	030523	020040															
	043352	051040	042510	030522															
	043360	000																	
10932	043361	040	020040	041520	DH14:	.ASCII / PC	TEST	FAILING	CONT.	CONT.	CONT.	CONT.	CONT.	/ <15><1					
	043366	020040	020040	052040															
	043374	051505	020124	043040															
	043402	044501	044514	043516															
	043410	020040	041440	047117															
	043416	027124	020040	041440															
	043424	047117	027124	020040															
	043432	041440	047117	027124															
	043440	020040	041440	047117															
	043446	027124	020040	041440															
	043454	047117	027124	006440															

10933	043462	012																				
	043463	040	020040	020040		.ASCIZ /		NO	REG	ADR	BAD	REG	RHC81	RHC82	RHD81	RHER1/						
	043470	020040	020040	020040																		
	043476	047516	020040	051040																		
	043504	043505	040440	051104																		
	043512	020040	040502	020104																		
	043520	042522	020107	051040																		
	043526	041510	030523	020040																		
	043534	051040	041510	031123																		
	043542	020040	051040	042110																		
	043550	030523	020040	051040																		
	043556	042510	030522	000																		
10934	043563	040	020040	041520	DH15:	.ASCIZ /	PC		REG,	ADR,/												
	043570	020040	051040	043505																		
	043576	020056	042101	027122																		
	043604	000																				
10935	043605	040	050040	020103	DH16:	.ASCII /	PC		WAT	BIT	REG				REG/<15><12>							
	043612	020040	020040	053440																		
	043620	052101	020040	020040																		
	043626	044502	020124	020040																		
	043634	020040	042522	020107																		
	043642	020040	020040	042522																		
	043650	006507	012																			
10936	043653	040	020040	020040		.ASCIZ /		PC	WANTED	ADDRESS	CONTENTS/											
	043660	020040	020040	050040																		
	043666	020103	020040	053440																		
	043674	047101	042524	020104																		
	043702	040440	042104	042522																		
	043710	051523	041440	047117																		
	043716	042524	052116	000123																		
10937	043724	020040	050040	020103	DH17:	.ASCII /	PC		TEST	CONT.	CONT.	CONT.	CONT.	CONT.	CONT.	CONT. /<15><12>						
	043732	020040	020040	042524																		
	043740	052123	020040	020040																		
	043746	047503	052116	020056																		
	043754	020040	047503	052116																		
	043762	020056	020040	047503																		
	043770	052116	020056	020040																		
	043776	047503	052116	020056																		
	044004	020040	047503	052116																		
	044012	020056	005015																			
10938	044016	020040	020040	020040		.ASCIZ /		NO	RHBA	RHDB	RHWC	RHC81	RHC82 /									
	044024	020040	020040	047040																		
	044032	020117	020040	020040																		
	044040	044122	040502	020040																		
	044046	020040	044122	041104																		
	044054	020040	020040	044122																		
	044062	041527	020040	020040																		
	044070	044122	051503	020061																		
	044076	020040	044122	051503																		
	044104	020062	000																			
10939	044107	040	020040	041520	DH20:	.ASCII /	PC		TEST	CONT	CONT	CONT	CONT	CONT	CONT /<15><12>							
	044114	020040	052040	051505																		
	044122	020124	020040	041440																		
	044130	047117	020124	020040																		

	044136	041440	047117	020124											
	044144	020040	041440	047117											
	044152	020124	020040	041440											
	044160	047117	020124	020040											
	044166	041440	047117	006524											
	044174	012													
10940	044175	040	020040	020040		.ASCIZ /	NO	RHER1	RHER2	RHER3	RHAS	RHDS1/			
	044202	020040	020040	047516											
	044210	020040	020040	051040											
	044216	042510	030522	020040											
	044224	051040	042510	031122											
	044232	020040	051040	042510											
	044240	031522	020040	051040											
	044246	040510	020123	020040											
	044254	051040	042110	030523											
	044262	000													
10941	044263	040	020040	041520	DH21:	.ASCII / PC	TEST	CONT	CONT	CONT	/<15><12>				
	044270	020040	020040	042524											
	044276	052123	020040	020040											
	044304	047503	052116	020040											
	044312	020040	047503	052116											
	044320	020040	020040	047503											
	044326	052116	020040	005015											
10942	044334	020040	020040	020040		.ASCIZ /	NO	RHCS1	RHAS	RHDS1 /					
	044342	020040	020040	047040											
	044350	020117	020040	020040											
	044356	044122	051503	020061											
	044364	020040	044122	051501											
	044372	020040	051040	042110											
	044400	030523	000040												
10943	044404	020040	041520	020040	DH22:	.ASCII / PC	TEST	RHAS	RHCS2	/<15><12>					
	044412	020040	042524	052123											
	044420	020040	020040	044122											
	044426	051501	020040	020040											
	044434	044122	051503	020062											
	044442	020040	005015												
10944	044446	020040	020040	020040		.ASCIZ /	NO	UNIT	UNIT/						
	044454	020040	047516	020040											
	044462	020040	020040	047125											
	044470	052111	020040	020040											
	044476	047125	052111	000											
10945	044503	120	020103	020040	DH24:	.ASCII /PC	RHDST	BAD	GOOD	SECTOR	SECTOR/<15><12>				
	044510	020040	051040	042110											
	044516	052123	020040	041040											
	044524	042101	020040	020040											
	044532	043440	047517	020104											
	044540	020040	051440	041505											
	044546	047524	020122	051440											
	044554	041505	047524	006522											
	044562	012													
10946	044563	040	020040	020040		.ASCIZ /	CONT.	RHLA	RHLA	NO	CLOCK/				
	044570	020040	041440	047117											
	044576	027124	020040	051040											
	044604	046110	020101	020040											

	044612	051040	046110	020101						
	044620	020040	047040	020117						
	044626	020040	020040	041440						
	044634	047514	045503	000						
10947	044641	120	020103	020040	DH26:	.ASCII	/PC	PC OF	FAILING	CONT. CONT. CONT./<15><12>
	044646	020040	050040	020103						
	044654	043117	020040	043040						
	044662	044501	044514	043516						
	044670	041440	047117	027124						
	044676	020040	020040	047503						
	044704	052116	020056	020040						
	044712	041440	047117	027124						
	044720	020040	020040	047503						
	044726	052116	006456	012						
10948	044733	040	020040	020040		.ASCIZ/		JSR	REG. ADDR	RHC81 RHC82 RHD81 RHER1/
	044740	020040	045040	051123						
	044746	020040	020040	051040						
	044754	043505	020056	042101						
	044762	051104	051040	041510						
	044770	030523	020040	051040						
	044776	041510	031123	020040						
	045004	051040	042110	030523						
	045012	020040	051040	042510						
	045020	030522	000							
10949	045023	120	020103	020040	DH27:	.ASCII	/PC	PC OF	TEST	FAILING GOOD RECEIVED/<15><12>
	045030	020040	050040	020103						
	045036	043117	020040	052040						
	045044	051505	020124	020040						
	045052	043040	044501	044514						
	045060	043516	043440	047517						
	045066	020104	020040	051040						
	045074	041505	044505	042526						
	045102	006504	012							
10950	045105	040	020040	020040		.ASCIZ /		JSR	NO	REGISTERDATA DATA/
	045112	020040	045040	051123						
	045120	020040	020040	020040						
	045126	047516	020040	020040						
	045134	051040	043505	051511						
	045142	042524	042122	052101						
	045150	020101	020040	042040						
	045156	052101	000101							
10951	045162	041520	020040	020040	DH30:	.ASCII	/PC	PC OF	REGISTER	GOOD BAD/<15><12>
	045170	020040	041520	047440						
	045176	020106	020040	042522						
	045204	044507	052123	051105						
	045212	043440	047517	020104						
	045220	020040	040502	006504						
	045226	012								
10952	045227	040	020040	020040		.ASCIZ /		JSR	ADDRESS	DATA DATA/
	045234	020040	045040	051123						
	045242	020040	020040	040440						
	045250	042104	042522	051523						
	045256	042040	052101	020101						
	045264	020040	040504	040524						

	046440	020040	051040	043505															
	046446	020056	020040	051040															
	046454	043505	020056	020040															
	046462	050040	051517	052111															
	046470	047117	051040	043505															
	046476	000056																	
10963	046500	041520	020040	020040	DH36:	.ASCII	/PC	PC OF	TEST	RHMR	POSITON	PATTERN/<15><12>							
	046506	020040	041520	047440															
	046514	020106	020040	042524															
	046522	052123	020040	020040															
	046530	044122	051115	020040															
	046536	020040	047520	044523															
	046544	047524	020116	040520															
	046552	052124	051105	006516															
	046560	012																	
10964	046561	040	020040	020040		.ASCIZ	/	JSR	NO	CONT.	REG.	REG./							
	046566	020040	020040	051512															
	046574	020122	020040	020040															
	046602	047516	020040	020040															
	046610	041440	047117	027124															
	046616	020040	051040	043505															
	046624	020056	020040	051040															
	046632	043505	000056																
10965	046636	041520	020040	020040	DH37:	.ASCII	/PC	TEST	POSITON	POSITON	GOOD	GOOD	PATTERN	DATA	N-CODE/				
	046644	020040	042524	052123															
	046652	020040	020040	047520															
	046660	044523	047524	020116															
	046666	047520	044523	047524															
	046674	020116	047507	042117															
	046702	020040	020040	047507															
	046710	042117	020040	020040															
	046716	040520	052124	051105															
	046724	020116	040504	040524															
	046732	020040	020040	026516															
	046740	047503	042504	005015															
10966	046746	020040	020040	020040		.ASCIZ	/	NO	ECC	GOOD	ECC1	ECC2	ECC	ENVELOPE	ZEROS/				
	046754	020040	047040	020117															
	046762	020040	020040	042440															
	046770	041503	020040	020040															
	046776	043440	047517	020104															
	047004	020040	042440	041503															
	047012	020061	020040	042440															
	047020	041503	020062	020040															
	047026	020040	041505	020103															
	047034	020040	047105	046126															
	047042	050117	020105	042532															
	047050	047522	000123																
10967						.EVEN													
10968	047054	001116	004174	024234	DT1:	.WORD		SERRPC, TSTNM, REGADR, SGDDAT, SBDDAT, 0											
	047062	001124	001126	000000															
10969	047070	001116	004174	031136	DT2:	.WORD		SERRPC, TSTNM, ERWORD, SGDDAT, 0											
	047076	001124	000000																
10970	047102	001116	004174	031136	DT3:	.WORD		SERRPC, TSTNM, ERWORD, SGDDAT, SBDDAT, 0											
	047110	001124	001126	000000															

10971	047116	001116	004174	001122	DT11:	.WORD	SERRPC, TSTNM, SBDADR, CS1, CS2, DS1, ER1, 0
	047124	001702	001700	001724			
	047132	001704	000000				
10972	047136	001116	004174	001122	DT14:	.WORD	SERRPC, TSTNM, SBDADR, SBDDAT, CS1, CS2, DS1, ER1, 0
	047144	001126	001702	001700			
	047152	001724	001704	000000			
10973	047160	001116	001172	000000	DT15:	.WORD	SERRPC, STMP1, 0
10974	047166	001116	001176	001172	DT16:	.WORD	SERRPC, STMP3, STMP1, STMP0, SBDDAT, 0
	047174	001170	001126	000000			
10975	047202	001116	004174	001676	DT17:	.WORD	SERRPC, TSTNM, BA, DB, WC, CS1, CS2, 0
	047210	001672	001674	001702			
	047216	001700	000000				
10976	047222	001116	004174	001704	DT20:	.WORD	SERRPC, TSTNM, ER1, ER2, ER3, AS, DS1, 0
	047230	001710	001716	001720			
	047236	001724	000000				
10977	047242	001116	004174	001702	DT21:	.WORD	SERRPC, TSTNM, CS1, AS, DS1, 0
	047250	001720	001724	000000			
10978	047256	001116	004174	001124	DT22:	.WORD	SERRPC, TSTNM, SGDDAT, SBDDAT, 0
	047264	001126	000000				
10979	047270	001116	001706	001126	DT24:	.WORD	SERRPC, DST, SBDDAT, STMP1, STMP2, STMP3, 0
	047276	001172	001174	001176			
	047304	000000					
10980	047306	001116	002002	001122	DT26:	.WORD	SERRPC, PCJSR, SBDADR, CS1, CS2, DS1, ER1, 0
	047314	001702	001700	001724			
	047322	001704	000000				
10981	047326	001116	002002	004174	DT27:	.WORD	SERRPC, PCJSR, TSTNM, REGADR, SGDDAT, SBDDAT, 0
	047334	024234	001124	001126			
	047342	000000					
10982	047344	001116	002002	024234	DT30:	.WORD	SERRPC, PCJSR, REGADR, SGDDAT, SBDDAT, 0
	047352	001124	001126	000000			
10983							
10984	047360	001116	004174	031136	DT31:	.WORD	SERRPC, TSTNM, ERWORD, SGDDAT, SBDDAT, CS1, DS1, ER1, 0
	047366	001124	001126	001702			
	047374	001724	001704	000000			
10985	047402	001116	002002	004174	DT32:	.WORD	SERRPC, PCJSR, TSTNM, ERWORD, SGDDAT, SBDDAT, CS1, DS1, ER1, 0
	047410	031136	001124	001126			
	047416	001702	001724	001704			
	047424	000000					
10986	047426	001116	002002	004174	DT33:	.WORD	SERRPC, PCJSR, TSTNM, ERWORD, SGDDAT, CS1, DS1, ER1, 0
	047434	031136	001124	001702			
	047442	001724	001704	000000			
10987	047450	001116	004174	027126	DT34:	.WORD	SERRPC, TSTNM, GECC1, GECC2, WECC1, WECC2, DISK, 0
	047456	027130	033734	033736			
	047464	032734	000000				
10988	047470	001116	004174	027126	DT35:	.WORD	SERRPC, TSTNM, GECC1, GECC2, EC2, EC1, POSITI, ER1, 0
	047476	027130	001734	001732			
	047504	027140	001704	000000			
10989	047512	001116	002002	004174	DT36:	.WORD	SERRPC, PCJSR, TSTNM, MR, EC1, EC2, 0
	047520	001722	001732	001734			
	047526	000000					
10990	047530	001116	004174	001732	DT37:	.WORD	SERRPC, TSTNM, EC1, POSITI, GECC1, GECC2, EC2, DATENV, ZCODE, 0
	047536	027140	027126	027130			
	047544	001734	027144	027146			
	047552	000000					

10991	047554	000	000	000	DF1:	.BYTE	0,0,0,0,0
	047557	000	000				
10992	047561	000	000	001	DF2:	.BYTE	0,0,1,0
	047564	000	000				
10993	047565	000	000	001	DF3:	.BYTE	0,0,1,0,0
	047570	000	000				
10994	047572	000	000	000	DF11:	.BYTE	0,0,0,0,0,0,0
	047575	000	000	000			
	047600	000	000				
10995	047601	000	000	000	DF14:	.BYTE	0,0,0,0,0,0,0,0
	047604	000	000	000			
	047607	000	000				
10996	047611	000	000		DF15:	.BYTE	0,0
10997	047613	000	000	000	DF16:	.BYTE	0,0,0,0
	047616	000	000				
10998	047617	000	000	000	DF17:	.BYTE	0,0,0,0,0,0,0
	047622	000	000	000			
	047625	000	000				
10999	047626	000	000	000	DF20:	.BYTE	0,0,0,0,0,0,0
	047631	000	000	000			
	047634	000	000				
11000	047635	000	000	000	DF21:	.BYTE	0,0,0,0,0
	047640	000	000				
11001	047642	000	000	000	DF22:	.BYTE	0,0,0,0
	047645	000	000				
11002	047646	000	000	000	DF24:	.BYTE	0,0,0,0,0,0
	047651	000	000	000			
11003	047654	000	000	000	DF26:	.BYTE	0,0,0,0,0,0,0
	047657	000	000	000			
	047662	000	000				
11004	047663	000	000	000	DF27:	.BYTE	0,0,0,0,0,0
	047666	000	000	000			
11005	047671	000	000	000	DF30:	.BYTE	0,0,0,0,0
	047674	000	000				
11006	047676	000	000	001	DF31:	.BYTE	0,0,1,0,0,0,0,0
	047701	000	000	000			
	047704	000	000				
11007	047706	000	000	000	DF32:	.BYTE	0,0,0,1,0,0,0,0,0
	047711	001	000	000			
	047714	000	000	000			
11008	047717	000	000	000	DF33:	.BYTE	0,0,0,1,0,0,0,0
	047722	001	000	000			
	047725	000	000				
11009	047727	000	000	000	DF34:	.BYTE	0,0,0,0,0,0,0
	047732	000	000	000			
	047735	000	000				
11010	047736	000	000	000	DF35:	.BYTE	0,0,0,0,0,0,0,0
	047741	000	000	000			
	047744	000	000				
11011	047746	000	000	000	DF36:	.BYTE	0,0,0,0,0,0
	047751	000	000	000			
11012	047754	000	000	000	DF37:	.BYTE	0,0,0,0,0,0,0,0,0
	047757	000	000	000			
	047762	000	000	000			

11013	047766	.EVEN
11014		
11015	000001	.END

ACL	=	000040	ACU	=	100000	AOE	=	001000	AS	=	001720
ATA	=	100000	ATABLE	=	004200	ATTENT	=	002004	ATO	=	000001
AT1	=	000002	AT2	=	000004	AT3	=	000010	AT4	=	000020
AT5	=	000040	AT6	=	000100	AT7	=	000200	A16	=	000400
A17	=	001000	BA	=	001676	BAI	=	000010	BASECH	=	030056
BEGIN	=	004220	BEGIN2	=	004210	BITST	=	024236	BIT0	=	000001
BIT00	=	000001	BIT01	=	000002	BIT02	=	000004	BIT03	=	000010
BIT04	=	000020	BIT05	=	000040	BIT06	=	000100	BIT07	=	000200
BIT08	=	000400	BIT09	=	001000	BIT1	=	000002	BIT10	=	002000
BIT11	=	004000	BIT12	=	010000	BIT13	=	020000	BIT14	=	040000
BIT15	=	100000	BIT2	=	000004	BIT3	=	000010	BIT4	=	000020
BITS	=	000040	BIT6	=	000100	BIT7	=	000200	BIT8	=	000400
BIT9	=	001000	BLT1	=	024264	BLT2	=	024306	BLT3	=	024316
BPTVEC	=	000014	CA	=	001714	CC	=	001740	CHECKD	=	104422
CHECKE	=	024706	CHECKT	=	024524	CLAREA	=	024406	CLDISK	=	024470
CLR	=	000040	COMHD	=	030706	COMPA	=	031730	COMPAR	=	025342
COMWHD	=	033776	COUNTD	=	032376	CRC	=	025646	CRLF	=	000200
CSF	=	000002	CSU	=	000010	CS1	=	001702	CS2	=	001700
CYL	=	031016	DAREAD	=	031072	DATENY	=	027144	DANORD	=	031076
DB	=	001672	DCK	=	100000	DCL	=	000100	DCLEAR	=	002026
DENVL	=	000200	DE1	=	000040	DFF20	=	000002	DF1	=	047554
DF11	=	047572	DF14	=	047601	DF15	=	047611	DF16	=	047613
DF17	=	047617	DF2	=	047561	DF20	=	047626	DF21	=	047635
DF22	=	047642	DF24	=	047646	DF26	=	047654	DF27	=	047663
DF3	=	047565	DF30	=	047671	DF31	=	047676	DF32	=	047706
DF33	=	047717	DF34	=	047727	DF35	=	047736	DF36	=	047746
DF37	=	047754	DFS	=	000001	DH1	=	042734	DH11	=	043202
DH14	=	043361	DH15	=	043563	DH16	=	043609	DH17	=	043724
DH2	=	043057	DH20	=	044107	DH21	=	044263	DH22	=	044404
DH24	=	044503	DH26	=	044641	DH27	=	045023	DH30	=	045162
DH31	=	045273	DH32	=	045475	DH33	=	045720	DH34	=	046131
DH35	=	046304	DH36	=	046500	DH37	=	046636	DIGB	=	000004
DISK	=	032734	DISPLA	=	177570	DLT	=	100000	DL64	=	000020
DMD	=	000001	DPR	=	000400	DRY	=	000200	DST	=	001706
DS1	=	001724	DT	=	001726	DTAGAP	=	033740	DTE	=	010000
DTSY	=	001000	DT1	=	047054	DT11	=	047116	DT14	=	047136
DT15	=	047160	DT16	=	047166	DT17	=	047202	DT2	=	047070
DT20	=	047222	DT21	=	047242	DT22	=	047256	DT24	=	047270
DT26	=	047306	DT27	=	047326	DT3	=	047102	DT30	=	047344
DT31	=	047360	DT32	=	047402	DT33	=	047426	DT34	=	047450
DT35	=	047470	DT36	=	047512	DT37	=	047530	DVA	=	004000
ECDATA	=	027124	ECH	=	000100	ECI	=	004000	ECORR	=	027550
ECTEST	=	027162	EC1	=	001732	EC2	=	001734	EMTVEC	=	000030
EM1	=	040130	EM11	=	040301	EM14	=	040334	EM15	=	040354
EM16	=	040425	EM17	=	040446	EM2	=	040213	EM20	=	040472
EM21	=	040513	EM22	=	040535	EM24	=	041130	EM25	=	041223
EM30	=	041263	EM31	=	041405	EM32	=	041530	EM33	=	042014
EM34	=	042066	EM35	=	042161	EM36	=	042460	EM37	=	042646
EM6	=	040242	ERCRC	=	034142	ERFLG8	=	001774	ERHDGP	=	034144
ERHEAD	=	034140	ERPOS	=	027546	ERR	=	040000	ERRVEC	=	000004
ERSTAR	=	030662	ERUNIT	=	030660	ERWORD	=	031136	ER1	=	001704
ER2	=	001710	ER3	=	001716	EXT1	=	000001	EXT10	=	000010
EXT2	=	000002	EXT20	=	000020	EXT4	=	000004	EXT40	=	000040
FEN	=	000200	FER	=	000020	FILLEC	=	027722	FIRST	=	004176

FMT22 = 010000	FNWORD 034162	FORMAT 032400	FOUT 034166
FSYNER 034136	FUTABL 002020	GCRC 034132	GECC1 027126
GECC2 027130	GO = 000001	GRV = 000010	HADTMP 027150
HARDER 027142	HCCRCE 026250	HCE = 000200	HCI = 002000
HCRC = 000400	HDESYN 034146	HDWSYN 032732	HEADER 032706
HEDGAP 031032	HEDSYN 031034	HEGAP 032720	HT = 000011
IAE = 002000	IE = 000100	ILF = 000001	ILLEGL 002062
ILR = 000002	IOTVEC= 000020	IR = 000100	IXE = 004000
KEY1 031022	KEY2 031024	KIPAR0= 172340	KIPAR1= 172342
KIPAR2= 172344	KIPAR3= 172346	KIPAR4= 172350	KIPAR5= 172352
KIPAR6= 172354	KIPAR7= 172356	KIPDR0= 172300	KIPDR1= 172302
KIPDR2= 172304	KIPDR3= 172306	KIPDR4= 172310	KIPDR5= 172312
KIPDR6= 172314	KIPDR7= 172316	LA 001736	LAD 024450
LERR 024232	LST = 002000	MAKECY 027004	MASK 024230
MCLK = 000002	MCPE = 020000	MHS = 001000	MID 026774
MIDDLE 026672	MINX = 000004	MMVEC = 000250	MOL = 010000
MPE = 000400	MR 001722	MRD = 000020	MSE = 000020
MSTCK = 000010	MWR = 000040	MXF = 001000	NCODE 027134
NCOUNT 027136	NED = 010000	NEM = 004000	NHS = 002000
NOPERA 002020	NOBYNC 031132	NOUNIT 001764	NOWORD 031064
NUNIT 001766	OCYL = 100000	OF 001712	OFREV = 000200
OFSETC 002052	OF100 = 000004	OF200 = 000010	OF25 = 000001
OF400 = 000020	OF50 = 000002	OF800 = 000040	OPERSE 023410
OPI = 020000	OR = 000200	OUT 031102	PAR = 000010
PAT = 000020	PC = 0000007	PCJBR 002002	PGE = 002000
PIE1 = 100000	PIE10 = 000100	PIE11 = 000040	PIE12 = 000020
PIE13 = 000010	PIE14 = 000004	PIE15 = 000002	PIE16 = 000001
PIE17 = 100000	PIE18 = 040000	PIE19 = 020000	PIE2 = 040000
PIE20 = 010000	PIE21 = 004000	PIE22 = 002000	PIE23 = 001000
PIE24 = 000400	PIE25 = 000200	PIE26 = 000100	PIE27 = 000040
PIE28 = 000020	PIE29 = 000010	PIE3 = 020000	PIE30 = 000004
PIE31 = 000002	PIE32 = 000001	PIE4 = 010000	PIE5 = 004000
PIE6 = 002000	PIE7 = 001000	PIE8 = 000400	PIE9 = 000200
PIP = 020000	PIRQ = 177772	PIRQVE= 000240	PKACK 002056
PLU = 020000	POSITI 027140	PRE = 000020	PROG = 001000
PR0 = 000000	PR1 = 000040	PR2 = 000100	PR3 = 000140
PR4 = 000200	PR5 = 000240	PR6 = 000300	PR7 = 000340
PS = 177776	PSEL = 002000	PSU = 000001	PSW = 177776
PUTREG 024170	PWRVEC= 000024	P12 027154	P22 027156
P24 027160	P3 027152	RCON 035304	RCYL 031122
RDCHR = 104412	RDHEAD 031140	RDLIN = 104414	RDOCT = 104416
RDY = 000200	READ 031732	READAT 002044	READIN 002060
RECALI 002024	REDATA 035306	REFOR 002046	REGADR 024234
REGSA1 037160	REINTO 003130	RELEAS 002030	RESVEC= 000010
RETCL 002054	RHAS 001650	RHBA 001626	RHCA 001644
RHCC 001670	RHCS1 001632	RHCS2 001630	RHDB 001622
RHDST 001636	RHDS1 001654	RHDT 001656	RHEC1 001662
RHEC2 001664	RHER1 001634	RHER2 001640	RHER3 001646
RHLA 001666	RHMR 001652	RHOF 001642	RHSN 001660
RHWC 001624	RKEY1 031126	RKEY2 031130	RMR = 000004
RNO 035302	RPTRP1 022702	RPTRP2 023006	RPVEC 001620
RPVECT 035416	RSETR 031124	RSYNC 031120	R0 = 0000000
R1 = 0000001	R2 = 0000002	R3 = 0000003	R4 = 0000004
R5 = 0000005	R6 = 0000006	R7 = 0000007	SAVDT 001776

SAVER	025146	SAVSN	002000	SC	= 100000	SCOP1	= 104420
SCRC	034214	SCYL	034204	SC1	= 000100	SC10	= 001000
SC2	= 000200	SC20	= 002000	SC4	= 000400	SEARCH	035032
SECGAP	032636	SECOTR	031020	SECTR	035030	SEECOM	002050
SEGPFR	034134	SELECT	001770	SERCH	002032	SETDSK	026122
SILOTB	032734	SKEY1	034210	SKEY2	034212	SKI	= 040000
SN	001730	SND1	005232	SP	= 0000006	SR0	= 177572
SR1	= 177574	SR2	= 177576	SR3	= 172516	SSECTR	034206
SSYN	031030	SS1	012066	SS10	012240	SS12	012270
SS13	012276	SS14	012304	SS15	012344	SS2	012514
SS3	012120	SS4	012124	SS5	012130	SS7	012156
STACK	= 001000	START	004224	STKLMT	= 177774	SWR	= 177570
SW0	= 000001	SW00	= 000001	SW01	= 000002	SW02	= 000004
SW03	= 000010	SW04	= 000020	SW05	= 000040	SW06	= 000100
SW07	= 000200	SW08	= 000400	SW09	= 001000	SW1	= 000002
SW10	= 002000	SW11	= 004000	SW12	= 010000	SW13	= 020000
SW14	= 040000	SW15	= 100000	SW2	= 000004	SW3	= 000010
SW4	= 000020	SW5	= 000040	SW6	= 000100	SW7	= 000200
SW8	= 000400	SW9	= 001000	TAGDTE	002016	TBITVE	= 000014
TDF	= 000040	TESDTE	002014	TESTAD	023406	TINCNT	025024
TKVEC	= 000060	TNPILL	002010	TOLGAP	033742	TOTALA	002006
TPVEC	= 000064	TRAPVE	= 000034	TRE	= 040000	TRK	030770
TRK1	= 004000	TRK10	= 040000	TRK2	= 010000	TRK20	= 100000
TRK4	= 020000	TRTVEC	= 000014	TSECC	002012	TSECCG	027132
TSTNM	004174	TST1	005340	TST10	010000	TST11	010026
TST12	010344	TST13	010774	TST14	011416	TST15	011770
TST16	012530	TST17	013142	TST2	005614	TST20	013644
TST21	014354	TST22	015102	TST23	015512	TST24	016214
TST25	016724	TST26	017452	TST27	020062	TST3	005660
TST30	020564	TST31	021304	TST32	022046	TST33	022610
TST34	022722	TST35	023016	TST4	005732	TST5	007036
TST6	007274	TST7	007636	TTNO	= 000034	TUF	= 000100
TY	031134	TYPDS	= 104410	TYPE	= 104400	TYPOC	= 104402
TYPON	= 104406	TYPOS	= 104404	T,SCOP	024452	UN	005650
UNIT	001762	UNIT8	001742	UNITSL	001772	UNLOAD	002022
UNS	= 040000	UPE	= 020000	US1	= 000001	US2	= 000002
US4	= 000004	UWR	= 000010	VUF	= 000002	VU30	= 010000
VV	= 000100	WAIT,T	025026	WAT	= 104424	WC	001674
WCE	= 040000	WCF	= 000040	WCRC	032716	WCU	= 000001
WCYL	034122	WECC1	033734	WECC2	033736	WKEY1	034126
WKEY2	034130	WLE	= 004000	WORD	031726	WRCHDA	025504
WRCHDT	002036	WRCHK	002034	WRCHHD	025200	WRDATA	032404
WRFROM	002064	WRHEAD	034216	WRIDAT	002040	WRIFOR	002042
WRITE	032164	WRL	= 004000	WRU	= 000400	WSECTR	034124
WSSYNC	032704	WSU	= 000004	WTRK	034060	WWORD	032162
X	031026	XE2	006552	Y	031066	ZCODE	027146
ZER	= 000400	ZWORDS	032402	SBDADR	001122	SBDAT	001126
SBELL	001210	SCHARC	036376	SCHTAG	001100	SCN1	= 000006
SCM2	= 000014	SCM3	= 000006	SCM4	= 000006	SCNTLC	037005
SCRLF	001215	SDBLK	036162	SDOAGN	023362	SDTBL	036152
SENDAD	023352	SENDCT	023302	SENDMG	023366	SENULL	023403
SEOP	023246	SEOPCT	023274	SERFLG	001103	SERMAX	001115
SERROR	037152	SERRPC	001116	SERRTD	001220	SERTY	037314
SERTTL	001112	SESCAP	001206	SFILLC	001150	SFILLS	001147

SGDADR	001120	SGDDAT	001124	SGET42	023324	SHD	=	000000
SHIOCT	037150	SICNT	001104	SILLUP	040112	SITEMB	001114	
SLF	001216	SLPADR	001106	SLPERR	001110	SMXCNT	035744	
SNUL	001146	SNWTST	000001	SOCNT	037712	SOMODE	037714	
SOVER	035730	SPASS	001100	SPOWER	040120	SPWRAD	040106	
SPWRDN	037764	SPWRMG	040102	SPWRUP	040032	SQUES	001214	
SRDCHR	036602	SRDLIN	036666	SRDOCT	037012	SRDSZ	=	000011
SREGAD	001152	SREG0	001154	SREG1	001156	SREG2	001160	
SREG3	001162	SREG4	001164	SREG5	001166	SRESET	023350	
SSAYR6	040116	SSCOPE	035460	SSETUP	=	000017		
SSTUP	=	SSVLAD	035702	SSWR	=	167700		
STIMES	001204	STKB	001140	STKCNT	036404	SSWRMK	=	000000
STKQEN	=	STKQIN	036406	STKQOU	036410	STKINT	036424	
STKS	001130	STKSRV	036474	STMP0	001170	STKQSR	036412	
STMP2	001174	STMP3	001176	STMP4	001200	STMP1	001172	
STN	=	STPB	001144	STPFLG	001151	STMPS	001202	
STRAP	037716	STRP	=	STRPAD	037736	STPS	001142	
STTYIN	036770	STYPDS	035746	STYPE	036172	STSTNM	001102	
STYPEX	036400	STYPOC	037514	STYPON	037530	STYPEC	036332	
STISTR	035506	STRP	=	STYPON	037530	STYPOS	037470	
				STFILL	037713	.	=	047766

ERRORS DETECTED: 0

*DERPTA, DERPTA_DERPTA, SNL, DERPTA.P11
 RUN-TIME: 72 01 1 SECONDS
 CORE USED: 29K