

# RK611

FUNCTIONAL CONTROLLER  
MD-11-DZR6K-B

EP-DZR6K-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 2

MADE IN USA

The image displays a dense grid of 100 small diagrams or tables, arranged in 10 rows and 10 columns. Each cell contains a small schematic or data table, likely representing functional blocks of a controller. The diagrams are too small to read individually but appear to be organized in a systematic grid.

# RK611

FUNCTIONAL CONTROLLER  
MD-11-DZR6K-B

EP-DZR6K-B-DL-A  
COPYRIGHT © 1976  
FICHE 2 OF 2

NOV 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The leftmost column of frames contains text, likely serving as a table of contents or index. The remaining frames contain data, which appears to be organized in a grid format, possibly representing a control schedule or a data table. The text and data are too small to be legible at this resolution.

B01

IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZRBK-B-D  
PRODUCT NAME: RK611 FUNCTIONAL CONTROLLER  
DIAGNOSTIC  
DATE: AUGUST, 1976  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: MARV TEGROTENHUIS

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE REQUIREMENTS
2.2	PRELIMINARY PROGRAMS
3.0	OPERATING PROCEDURE
3.1	LOADING PROCEDURE
3.2	STARTUP PROCEDURE
3.3	CONSOLE SWITCH REGISTER
3.4	SOFTWARE SWITCH REGISTERS
3.5	UNIBUS ADDRESS
3.6	EXECUTION TIME
4.0	PROGRAM DESCRIPTION
5.0	ERROR REPORTING
6.0	SUBROUTINES

## 1.0 ABSTRACT

THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC (DZR6K) IS A SERIES OF TESTS THAT COMPLETES THE TESTING OF AN RK611/RK06 SUBSYSTEM. THE DISKLESS CONTROLLER DIAGNOSTIC AND THE RK06 DRIVE DIAGNOSTIC ARE PREREQUISITES TO THE RUNNING OF THIS PROGRAM. THE PURPOSE OF THIS PROGRAM IS TO TEST THOSE AREAS IN THE CONTROLLER THAT COULD NOT BE TESTED IN A DISKLESS ENVIRONMENT AND THOSE AREAS OF THE DRIVE THAT COULD NOT BE TESTED UNTIL CONTROLLER OPERATION IN A DIAGNOSTIC OR MAINTENANCE MODE HAS BEEN TESTED.

THE TESTS PERFORMED ARE MAINLY FUNCTIONALLY ORIENTED BUT DIAGNOSTIC MODE IS USED IN NUMEROUS OCCASSIONS TO ACCOMPLISH THE OBJECTIVES, MAINLY THE FORCING OF ERRORS. IN THESE CASES, THE CONTROLLER IS PLACED IN DIAGNOSTIC MODE AND OPERATION IS CLOCKED PART WAY THROUGH. DIAGNOSTIC MODE IS THEN RESET AND THE CONTROLLER ALLOWED TO COMPLETE THE OPERATION. DEPENDING ON THE OPERATION AND HOW FAR THROUGH IT BEFORE DIAGNOSTIC MODE IS RESET VARIOUS ERROR CONDITIONS CAN BE MADE TO OCCUR. THIS DOCUMENT DOES NOT ATTEMPT TO EXPLAIN WHY THESE ERROR CONDITIONS ARE SET BUT THE INDIVIDUAL TEST DESCRIPTIONS SPECIFY WHAT ERROR IS BEING FORCED AND THE PROCEDURE USED TO FORCE IT.

## CAUTION

THIS PROGRAM SHOULD BE HALTED ONLY AT END OF PASS. IF THE PROGRAM IS HALTED AT ANY OTHER POINT THE POSSIBILITY EXISTS THAT THE CARTRIDGE FORMAT WILL BE INCORRECT, THE CYLINDER ADDRESS IN THE DRIVE MAY BE INVALID, OR THE DRIVE MAY NOT BE READY.

## 2.0 REQUIREMENTS

## 2.1 HARDWARE REQUIREMENTS

PDP-11 SYSTEM (16K MEMORY)  
CONSOLE TERMINAL  
DECTAPE, PAPERTAPE, OR DISK  
LINE CLOCK (KW11-L) (OPTIONAL)  
PARITY OPTION (MM11) (OPTIONAL)  
RK611 CONTROLLER  
AT LEAST 1 AND UP TO 8 RK06 DRIVES  
FORMATTED RK06K ON EACH DRIVE

## 2.2 PRELIMINARY PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC (ALL PARTS) AND THE UNIBUS RK06 DRIVE DIAGNOSTIC (ALL PARTS) SHOULD HAVE RUN SUCCESSFULLY.

### 3.0 OPERATING PROCEDURE

#### 3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM BINARY TAPE USING THE ABSOLUTE LOADER OR FROM XXDP MEDIA SUPPORTED BY XXDP.

IT CAN BE LOADED AND RUN UNDER APT OR ACT AND IT CAN BE CHAINED BY XXDP.

#### 3.2 STARTUP PROCEDURE

THE PROGRAM START LOCATION IS 200(8). THIS START WILL AUTOMATICALLY SIZE THE SYSTEM UNLESS RUNNING UNDER APT. THE PROGRAM ASSUMES THE STANDARD UNIBUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY LEVEL (177440, 210, AND 4 RESPECTIVELY). IF STARTED AT 200 AND THE XXDP MEDIA IS RK06 (PROGRAM LOADED FROM RK06) DRIVE 0 IS NOT TESTED.

LOCATION 204(8) IS THE PROGRAM RESTART.

LOCATION 214(8) IS THE PARAMETERIZATION START LOCATION. THE OPERATOR WILL BE ASKED TO IDENTIFY THE BUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY. IF THE PROGRAM WAS LOADED FROM RK06, THE OPERATOR WILL BE ASKED TO MOUNT A WORK CARTRIDGE ON DRIVE 0 OR TO PLACE IT OFF-LINE IF IT IS NOT TO BE TESTED.

LOCATION 220(8) IS THE PHASE LOCKED LOOP CLOCK ADJUSTMENT START. THE PROGRAM FIRST RUNS THE FIRST THREE TESTS AND THEN JUMPS TO THE ADJUSTMENT ROUTINE. THE PROGRAM WILL CONTINUE TO LOOP IN THIS ROUTINE UNTIL THE PROCESSOR IS HALTED.

ALL DRIVES THAT ARE TO BE TESTED MUST BE ON-LINE, READY, AND WRITE LOCK RESET. IF ALL THREE CONDITIONS ARE NOT MET, THAT DRIVE IS NOT TESTED.

#### 3.3 CONSOLE SWITCH REGISTER

THE CONSOLE SWITCH REGISTER IS USED TO PROVIDE PROGRAM CONTROL AS DESCRIBED BELOW:

- SW15 - HALT ON ERROR
- SW14 - LOOP ON TEST
- SW13 - INHIBIT ERROR REPORT
- SW12 - ABORT PROGRAM AFTER 20 ERRORS
- SW11 - INHIBIT ITERATIONS
- SW10 - BELL ON ERROR
- SW09 - LOOP ON ERROR
- SW08 - EXECUTE TEST NUMBER SPECIFIED IN SW<7-0>.
- SW<7-0> - EXECUTE THIS TEST IF SW08 SET.

EXECUTING A SPECIFIC TEST MUST BE USED WITH CAUTION. SOME TESTS REQUIRE OTHERS TO BE RUN TO FORMAT THE PACK IN A SPECIFIC MANNER OR WRITE SPECIFIC DATA. TESTS THAT REQUIRE OTHERS TO BE RUN INDICATE THIS IN THE TEST DESCRIPTION. IT IS

SUGGESTED THAT THE PROGRAM BE RUN IN THE DEFAULT SEQUENCE THE FIRST TIME AFTER IT HAS BEEN LOADED.

NOTE: TEST 3 MUST BE RUN BEFORE ANY SUBSEQUENT TEST. THIS TEST DETERMINES WHICH DRIVES ARE ON THE DRIVE BUS FOR ALL FOLLOWING TESTS. LIKEWISE, TEST 20 MUST BE RUN BEFORE ANY TEST SUBSEQUENT TO IT. THIS TEST READS THE BAD SECTOR FILES AND BUILDS TABLES USED BY THE FOLLOWING TESTS. THESE TESTS, HAVING BEEN RUN ONCE, NEED NOT BE RUN AGAIN IF A DIFFERENT TEST IS SELECTED.

### 3.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E., AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RKO6 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW ='

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

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

### 3.5 UNIBUS ADDRESSES

STANDARD UNIBUS ADDRESSES ARE ASSUMED FOR THE KW11-L AND MM11 OPTIONS. THESE ADDRESSES MAY BE CHANGED BY CHANGE THE APPROPRIATE MEMORY LOCATIONS. THE FOLLOWING TAGS AND LOCATIONS HAVE BEEN USED:

	TAG	LOCATION
KW11-L		
UNIBUS ADDRESS	KWLADD	1674
VECTOR ADDRESS	KLVEC	1676
MM11		
UNIBUS ADDRESS		
BANK 0	MMCSR1	1700
BANK 1	MMCSR2	1702
VECTOR ADDRESS	MMVEC	1704

### 3.6 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM FOR ONE DRIVE IS APPROXIMATELY 50 SECONDS AND EACH SUBSEQUENT PASS IS APPROXIMATELY 6 MINUTES 50 SECONDS.

THE EXECUTION TIME FOR MULTIPLE DRIVES IS THE PRODUCT OF THE NUMBER OF DRIVES TIMES 52 SECONDS FOR THE FIRST PASS. FOR SUBSEQUENT PASSES THE RUN TIME IS THE PRODUCT OF 6 MINUTES 50 SECONDS TIMES THE NUMBER OF DRIVES PLUS 25 SECONDS FOR EACH DRIVE AFTER THE FIRST.

4.0 PROGRAM DESCRIPTION

THE FOLLOWING TEST SEQUENCE IS EXECUTED ASSUMING TWO OR MORE DRIVES.

FIRST PASS - FIRST DRIVE:  
ALL TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE.

FIRST PASS - ALL REMAINING DRIVES:  
STATUS VALID TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH DRIVE.

THEN MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH COMBINATION OF DRIVES.

SECOND AND ALL SUBSEQUENT PASSES:  
THE SAME SEQUENCE OF TESTING IS REPEATED EXCEPT FOR TEST ITERATIONS WHICH ARE SPECIFIED FOR EACH TEST.

\*\*\*\*\*  
\*\*BASIC INTERFACE AND OPTION TESTS  
\*\*\*\*\*

TEST 1 RK611 BASE ADDRESS TEST

CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT CAUSE A NON-EXISTANT MEMORY TRAP.

TEST 2 INTERRUPT VECTOR ADDRESS TEST CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE EXPECTED ADDRESS.

\*\*\*\*\*  
\*\*STATUS VALID TESTS  
\*\*\*\*\*

TEST 3 SELECT ALL DRIVES

IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT, DETERMINE WHAT DRIVES ARE ON-LINE BY SELECTING ALL DRIVES. IF NON-EXISTENT DRIVE REPORTED MAKE SURE STATUS VALID IS RESET. IF DRIVE PRESENT MAKE SURE NO ERROR EXISTS, DRIVE IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.



IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE CYCLED UP, AND STATUS VALID SET.

IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM START (214) WAS USED. IF THE AUTOMATIC START (200) IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204) WILL RETAIN THE TEST STATUS OF DRIVE 0.

IF THE PARAM START IS USED, THE OPERATOR MUST EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.

THE DRIVE MUST BE ON-LINE, CYCLED UP, AND WRITE ENABLED. IF ANY ONE OF THESE CONDITIONS IS NOT TRUE THAT DRIVE IS NOT TESTED AND IT IS EXPECTED TO BE OFF-LINE. ADDRESSING THAT DRIVE SHOULD CAUSE NON-EXISTANT DRIVE ERROR. AT COMPLETION OF THE TEST A MESSAGE WILL BE GIVEN TO IDENTIFY THE DRIVES TO BE USED IN TESTING.

NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE ANY OTHER TEST THAT FOLLOWS.

TEST 4 RELEASE ALL DRIVES

RELEASE ALL DRIVES. MAKE SURE NO ERROR SETS AND STATUS VALID IS RESET.

TEST 5 NON-STANDARD MESSAGES AND SVAL

PICK ONE OF THE AVAILABLE DRIVES AND GET NON-STANDARD MESSAGES. MAKE SURE NO ERROR OCCURS AND STATUS VALID DOES NOT SET AND THAT NON-STANDARD MESSAGES CAUSE STATUS VALID TO RESET.

TEST 6 WRITING CS2 AND STATUS VALID

SELECT AN AVAILABLE DRIVE. MAKE SURE STATUS VALID IS SET. WRITE COMMAND AND STATUS REGISTER 2. MAKE SURE STATUS VALID RESETS.

\*\*\*\*\*  
\*\*CONTROLLER ERROR TESTS  
\*\*\*\*\*

TEST 7 DRIVE TYPE ERROR

CREATE A DRIVE TYPE ERROR MAKE SURE DRIVE TYPE ERROR SETS AND STATUS VALID SETS.

## TEST 10 STATUS VALID AND PARITY ERROR

ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY. MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION, DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT, AND STATUS VALID SET. ISSUE A CONTROLLER CLEAR. MAKE SURE DRIVE INTERRUPT AND ATTENTION ARE STILL SET. SELECT DRIVE AGAIN WITH GOOD PARITY. MAKE SURE ATTENTION, DRIVE STATUS CHANGE, DRPAR, CONTROLLER ERROR, DRIVE INTERRUPT, AND STATUS VALID ARE SET AND SPAR IS RESET. ISSUE A CONTROLLER CLEAR. GET NON-STANDARD MESSAGES AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION ARE SET. CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT FOR ALL AVAILIABLE DRIVES.

## TEST 11 UNIT FIELD ERROR ON RELEASE

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A RELEASE COMMAND. CLOCK THROUGH PHASE ADDRESS 2. TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD ERROR SETS.

## TEST 12 UNIT FIELD ERROR ON SELECT

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE SELECTED = 0. CLOCK THROUGH PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD ERROR SETS.

\*\*\*\*\*  
 \*\*ATTENTION HANDLING BY CONTROLLER  
 \*\*\*\*\*

## TEST 13 DOUBLE INTERRUPT

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. MAKE SURE STATUS VALID IS SET. CHECK THAT SECOND INTERRUPT OCCURS. AFTER SECOND INTERRUPT CHECK THAT STATUS VALID IS RESET. ISSUE SELECT AND MAKE SURE STATUS VALID SETS. CLEAR DRIVE. CHECK THAT DRIVE STATUS CHANGE SETS (BIT 14 OF DRIVE STATUS REGISTER)

## TEST 14 SINGLE INTERRUPT FROM ATTENTION

DO A SEEK TO CYLINDER 0. WAIT FOR INTERRUPT FROM DRIVE ATTENTION. LOWER PRIORITY AGAIN AND MAKE SURE ANOTHER INTERRUPT DOES NOT OCCUR. CLEAR DRIVE.

## TEST 15 RESET ATTENTIONS WITH UNIBUS INIT

DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES. ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.

\*\*\*\*\*  
\*\*ILLEGAL DISK ADDRESS ERROR TESTS  
\*\*\*\*\*

TEST 16 ILLEGAL DISK ADDRESS (PART 1)

ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS. CLEAR CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7, CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.

TEST 17 ILLEGAL DISK ADDRESS (PART 2)

ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE ILLEGAL DISK ADDRESS ERROR SETS. CLEAR CONTROLLER AND DRIVE

\*\*\*\*\*  
\*\*WRITE HEADER TESTS  
\*\*\*\*\*

TEST 20 READ BAD SECTOR INFORMATION

ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632, TRACK 2 TO GET THE FACTORY DETECTED BAD SECTOR FILE, 26 SECTOR MODE.

IF AN ERROR OCCURS, READ SECTOR 2, 4, 6, OR 10(8) UNTIL A SUCCESSFUL READ IS DONE. IF NONE READ SUCCESSFULLY REMOVE THIS DRIVE FROM TEST. WHEN A READ IS SUCCESSFUL, TEST THAT THE PACK IS NOT AN ALIGNMENT PADK AND STORE THE ENTRIES FOR LATER USE.

REPEAT THIS SERIES OF OPERATIONS FOR FACTORY DETECTED BAD SECTORS 24 SECTOR MODE, SOFTWARE DETECTED BAD SECTORS 26 SECTOR MODE, AND SOFTWARE DETECTED BAD SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR 24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED FROM TESTING.

NOTE: THIS TEST IS RUN IN THE FIRST (QUICK VERIFY) PASS ONLY.

TEST 21 FORMAT IN 26 SECTOR FORMAT

FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER AFTER READ HEADER.

\*\*\*\*\*  
\*\*HEADER RECOGNITION TESTS  
\*\*\*\*\*

## TEST 22 BAD SECTOR ERROR

FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1 (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS AND ALL OTHER SECTORS GOOD.

ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS. ISSUE A WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS. MAKE SURE BAD SECTOR ERROR SET. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2. MAKE SURE NO ERROR SETS.

## TEST 23 HEADER VRC ERROR

FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE 16 SECTORS WITH BAD HEADER VRC. ISSUE A WRITE DATA OF EACH OF THE SECTORS WITH A BAD HEADER VRC. MAKE SURE HEADER VRC ERROR SETS. ISSUE A WRITE DATA TO A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.

## TEST 24 BAD SECTOR ERROR AND HVRC ERROR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS BOTH A BAD SECTOR ERROR AND HEADER VRC. ISSUE A WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE ONLY HEADER VRC ERROR SETS.

## TEST 25 OPERATION INCOMPLETE

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE WRONG FORMAT. ISSUE A WRITE DATA OF 400 TO CYLINDER 0, TRACK 0, SECTOR 21. MAKE SURE OPI SET.

## TEST 26 OPI WITH HVRC ERROR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0, SECTOR 17. THAT BOTH OPERATION INCOMPLETE AND HEADER VRC SET.

## TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0, AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE END OF THE OPERATION.

\*\*\*\*\*  
\*\*DATA TRANSFER TESTS  
\*\*\*\*\*

TEST 30 WRITE AND READ ONE SECTOR

FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313,  
TRACK 2 TO AGREE WITH BAD SECTOR INFORMATION. ISSUE  
A WRITE DATA OF ONE SECTOR ON CYLINDER 312, TRACK 0.  
READ IT BACK TO MAKE SURE IT AGREES WITH WHAT IS  
WRITTEN.

TEST 31 WRITE DATA ADDRESS GREATER THAN 32K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.  
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K OF  
MEMORY IS PRESENT.

TEST 32 READ DATA ADDRESS GREATER THAN 32K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 177770.  
CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K OF  
MEMORY IS PRESENT.

TEST 33 WRITE DATA ADDRESS GREATER THAN 64K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.  
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K OF  
MEMORY IS PRESENT.

TEST 34 READ DATA ADDRESS GREATER THAN 64K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.  
CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K OF  
MEMORY IS PRESENT.

TEST 35 WRITE DATA ADDRESS GREATER THAN 96K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.  
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K OF  
MEMORY IS PRESENT.

TEST 36 READ DATA ADDRESS GREATER THAN 96K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.

CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K OF MEMORY IS PRESENT.

TEST 37 PARTIAL SECTOR WRITE DATA

ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312, HEAD 0, SECTOR 0. MAKE SURE THE SECTOR WAS ZERO FILLED CORRECTLY.

TEST 40 PARTIAL SECTOR READ DATA

WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A KNOWN CONFIGURATION. ISSUE A READ DATA OF 103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE ONLY 103 WORDS GET TRANSFERRED TO MEMORY.

TEST 41 WRITE DATA WITH NON-EXISTENT MEMORY

ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000. MAKE SURE NON-EXISTENT MEMORY SETS.

TEST 42 READ DATA WITH NON-EXISTENT MEMORY

ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000. MAKE SURE NON-EXISTENT MEMORY SETS.

TEST 43 UNIBUS PARITY ERROR

INITIALIZE A MEMORY LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF 400 WORDS STARTING AT A LOCATION 110 WORDS BEFORE THE LOCATION WITH BAD PARITY. MAKE SURE THAT UNIBUS PARITY ERROR SETS.

NOTE: THIS TEST IS ONLY EXECUTED IF MEMORY PARITY OPTION EXISTS FOR BUFFER.

\*\*\*\*\*  
\*\*MULTIPLE SECTOR OPERATIONS  
\*\*\*\*\*

TEST 44 TWO SECTOR WRITE DATA (PART 1)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE IT IS CORRECT.

TEST 45 TWO SECTOR WRITE DATA (PART 2)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 23. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID NOT TAKE PLACE.

## TEST 46 TWO SECTOR WRITE DATA (PART 3)

ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312, TRACK 0, SECTOR 10. READ DATA BACK ONE SECTOR AT A TIME AND CHECK ZERO FILL OF SECOND SECTOR.

## TEST 47 MID-TRANSFER SEEK ON WRITE (PART 1)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 25. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

## TEST 50 MID-TRANSFER SEEK ON WRITE (PART 2)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 2, SECTOR 25. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

## TEST 51 TWO SECTOR READ DATA (PART 1)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS READ.

NOTE: TWO SECTOR WRITE DATA (PART 1) MUST BE EXECUTED BEFORE THIS TEST.

## TEST 52 TWO SECTOR READ DATA (PART 2)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 23. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.

NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE EXECUTED BEFORE THIS TEST.

## TEST 53 TWO SECTOR READ DATA (PART 3)

ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312, TRACK 0, SECTOR 10. VERIFY THAT ALL 401 WORDS ARE PLACED IN MEMORY.

NOTE: TWO SECTOR WRITE DATA (PART 3) MUST BE EXECUTED BEFORE THIS TEST.

## TEST 54 MID-TRANSFER SEEK ON READ (PART 1)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 25. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES OCCUR.

NOTE: MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE EXECUTED BEFORE THIS TEST.

## TEST 55 MID-TRANSFER SEEK ON READ (PART 2)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 2, SECTOR 25. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES OCCUR.

NOTE: MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE EXECUTED BEFORE THIS TEST.

TEST 56 CYLINDER ADDRESS OVERFLOW (PART 1)

ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632, TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS OVERFLOW ERROR DOES NOT OCCUR.

TEST 57 CYLINDER ADDRESS OVERFLOW (PART 2)

ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632, TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS OVERFLOW ERROR DOES OCCUR.

\*\*\*\*\*  
 \*\*18 BIT DATA TRANSFER TESTS  
 \*\*\*\*\*

TEST 60 FORMAT IN 24 SECTOR FORMAT

FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER AFTER READ HEADER.

TEST 61 24 SECTOR FORMAT DATA TRANSFER (PART 1)

ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0, SECTOR 0. READ SECTOR BACK AND MAKE SURE IT IS CORRECT.

TEST 62 24 SECTOR FORMAT DATA TRANSFER (PART 2)

LOAD A LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0, SECTOR 0 WITH BUFFER BEGINNING 110 WORDS BEFORE WORD WITH BAD PARITY. MAKE SURE UNIBUS PARITY ERROR DOES NOT SET. READ SECTOR BACK AND MAKE SURE IT IS CORRECT.

NOTE: THIS TEST IS EXECUTED ONLY IF MEMORY PARITY EXISTS FOR SPECIFIED LOCATION.

TEST 63 24 SECTOR FORMAT DATA TRANSFER (PART 3)

ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0, SECTOR 23. READ SECTOR BACK AND MAKE SURE IT IS CORRECT. MAKE SURE THAT MID-TRANSFER SEEK HAS TAKEN PLACE.



\*\*\*\*\*  
 \*\*SPECIAL DATA TRANSFER TESTS  
 \*\*\*\*\*

TEST 64 MULTI-SECTOR DATA TRANSFER AND BSE

FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH SECTOR 1 MARKED BAD. ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS AND RKDA IS CORRECT. READ SECTOR 0 AND MAKE SURE IT IS CORRECT.

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS AND THE PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.

TEST 65 FORMAT TEST

FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT. MAKE SURE NO ERRORS SET. READ SECTORS 0-25 AND MAKE SURE DATA CHECK DOES NOT OCCUR.

\*\*\*\*\*  
 \*\*WRITE CHECK TESTS  
 \*\*\*\*\*

TEST 66 WRITE-CHECK WITH NO ERROR

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN. DO A WRITE-CHECK OF 400 WORDS. MAKE SURE NO ERROR OCCURS.

TEST 67 WRITE CHECK ERROR (PART 1)

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING CONFIGURATIONS:

000001 000020 000400 010000  
 000002 000040 001000 020000  
 000004 000100 002000 040000  
 000010 000200 004000 100000

MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT IS CORRECT.

TEST 70 WRITE CHECK ERROR (PART 2)

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777 IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS ONE OF THE FOLLOWING CONFIGURATIONS:

177776 177757 177377 167777  
 177775 177737 176777 157777  
 177773 177677 175777 137777  
 177767 177577 173777 077777

MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT IS CORRECT.

#### TEST 71 WRITE CHECK OF PARTIAL SECTOR

WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF 110 WORDS MAKING SURE THE 111TH WORD IS DIFFERENT THAN DATA ON DISK. MAKE SURE WRITE CHECK ERROR DOES NOT SET.

\*\*\*\*\*  
 \*\*MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT  
 \*\*\*\*\*

#### TEST 72 MAXIMUM DATA TRANSFER (PART 1)

IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO INSURE THE FORMAT IS CORRECT.

ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH ONE SECTOR WRITES. ISSUE A SEEK TO CYLINDER 0, TRACK 0. ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 20000 WORDS TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE CONTROLLER TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS, DISK ADDRESS, BUS ADDRESS AND WORD COUNT. READ EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.

NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT IN THE FIRST 256 SECTORS ON THE PACK.

#### TEST 73 MAXIMUM DATA TRANSFER (PART 2)

ZERO OUT FIRST 256 SECTORS OF THE DISK WITH 20000 WORD WRITE. SEEK TO CYLINDER 632. ISSUE A WRITE OF MAXIMUM DATA TRANSFER 20000 WORD WRITE. MAKE SURE CONTROLLER TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS DISK ADDRESS, BUS ADDRESS AND WORD COUNT. SEEK TO CYLINDER 632. ISSUE A WRITE CHECK OF 20000 WORDS. MAKE SURE NO ERROR SETS.

NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT IN THE FIRST 256 SECTORS ON THE PACK.

#### TEST 74 CONTROLLER TIME OUT

SEEK TO CYLINDER 632. ISSUE A RECALIBRATE AND DO NOT WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER OF CYLINDER 0, TRACK 0. MAKE SURE CONTROLLER TIME OUT SETS.

\*\*\*\*\*  
 \*\*ERRORS DURING DATA TRANSFER  
 \*\*\*\*\*

TEST 75 LIMIT DETECT ON DATA TRANSFER

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE A SEEK TO CYLINDER 2 WITH BAD PARITY. ISSUE A DRIVE CLEAR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1, TRACK 0, HEAD 0. SEEK INCOMPLETE BECAUSE OF OUTER

TEST 76 PROGRAMMING ERROR

ISSUE A SUBSYSTEM CLEAR. ISSUE A READ DATA OF 400 WORDS ON CYLINDER 0, TRACK 0, SECTOR 0. DURING READ ISSUE A WRITE TO THE SPARE REGISTER. MAKE SURE PROGRAMMING ERROR SETS.

TEST 77 ECC HARD

ISSUE A SUBSYSTEM CLEAR. ISSUE A WRITE DATA WORDS CONSISTING OF 177777 TO CYLINDER 0, TRACK 0, SECTOR 0. NOW WRITE ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0. DURING WRITE ISSUE CONTROLLER CLEAR. MAKE SURE PROGRAMMING ERROR IS RESET. NOW ISSUE A READ DATA TO CYLINDER 0, TRACK 0, HEAD 0 AND AN ECC HARD ERROR SHOULD SET.

TEST 100 DRIVE TIMING ERROR

ISSUE A SUBSYSTEM CLEAR. SEEK TO CYLINDER 632. ISSUE A RECALIBRATE BUT DO NOT WAIT FOR SECOND INTERRUPT. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER OF CYLINDER 0, TRACK 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE TIMING ERROR SHOULD SET BECAUSE OF NO DATA TRANSITIONS ON DATA LINE.

\*\*\*\*\*  
 \*\*ERROR FORCING IN DRIVE  
 \*\*\*\*\*

TEST 101 INITIALIZE CLEARING SACK

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE. ISSUE A SUBSYSTEM CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE SELECTED = 0. CLOCK THROUGH PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD ERROR DOES NOT SET.

## TEST 102 DRIVE OFF TRACK

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0, TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILIABLE DRIVES.

## TEST 103 FILE UNSAFE

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECLAIBRATE. ISSUE A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR FORMAT. DO A SELECT COMMAND IN 26 SECTOR FORMAT. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR FORMAT. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE INDEX PULSE. TURN OFF DIAGNOSTIC MODE. FILE UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE THROUGH SECTOR PULSE. REPEAT FOR ALL AVAILIABLE DRIVES.

## TEST 104 DUMMY TEST FOR PREVIOUS TEST EXIT

THIS TEST IS PRESENT TO MAKE \$SWOBTB TABLE HAVE AN ENTRY WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST. IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".

IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS FROM THE DRIVE BECOMING READY AT A LATER TIME.

\*\*\*\*\*  
 \*\*MULTI-DRIVE OPERATIONS  
 \*\*\*\*\*

## TEST 105 RESET ATTENTIONS WITH UNIBUS INIT

DO A RECALIBRATE ON ALL AVAILIABLE DRIVES. ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.

## TEST 106 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

DO A RECALIBRATE ON ALL AVAILIABLE DRIVES. ISSUE A SUBSYSTEM CLEAR. MAKE SURE ALL ATTENTIONS RESET.

## TEST 107 SVAL AND ATTENTION FROM OTHER DRIVE

DO A RECALIBRATE ON ONE AVAILIABLE DRIVE. DO A SELECT ON ANOTHER AVAILIABLE DRIVE. MAKE SURE STATUS VALID IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE MAKE

SURE STATUS VALID REMAINS SET AND DRIVE STATUS CHANGE REMAINS RESET.

REPEAT FOR ALL COMBINATIONS OF TWO AVAILABLE DRIVES.

NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST TWO DRIVES ARE AVAILABLE.

#### TEST 110 OVERLAPPED OPERATIONS

DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B. AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD BE SET AND DRIVE A HAS ATTENTION SET.

REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.

NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE TEST WILL NOT BE DONE.

#### 5.0 ERROR REPORTING

A DETAILED DESCRIPTION OF THE ERROR FORMATS AND REPORTS CONTENTS IS GIVEN HERE. THIS IS ESSENTIALLY THE SAME AS CAN BE FOUND IN THE LISTING COMMENTS UNDER THE ERROR POINTER TABLE.

##### ERROR POINTER TABLE:

THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR. THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.

NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).

NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS

EXPLAINED AS FOLLOWS:

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

EM AND DH ARE ASCIZ DATA. EM IS ALWAYS A MESSAGE BUT DH CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACCROSS THE PAGE. DT IS A STRING OF WORDS THAT POINT TO THE DATA TO BE TYPED, AND DF IS A STRING OF WORDS THAT TELL HOW THE DT WORDS ARE TO BE TYPED. IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A PARTICULAR FORMAT, IT IS REPLACED WITH A ZERO. THE NORMAL USAGE OF THE ERROR TABLE IS TO HAVE A TABLE ENTRY FOR EACH ERROR MESSAGE THAT CAN OCCUR. IN THE INTEREST OF ECONOMICS OF CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE IN A SLIGHTLY DIFFERENT MANNER AS DESCRIBED BELOW.

THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT FORMATS THAT ARE USED THROUGHOUT THE PROGRAM. WHEN AN ERROR IS TO BE REPORTED, THE TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT IS

CHOSEN FROM THE DEFINED SET. THE TABLE ENTRY CHOSEN IS THEN ALTERED BY CHANGING THE FIRST (AND POSSIBLY THE SECOND) WORD TO CONTAIN THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE MESSAGE PORTION OF THE REPORT. THE DATA FIELDS FOR THAT ENTRY ARE NEVER CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.

THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW WITH THE DEFINITION OF THE ENTRY. ALL DATA FIELDS ARE TYPED IN OCTAL.

ERROR ITEM 1  
(MESSAGE)  
TST NUM ERR PC DRIVE  
STESTN SERRPC DRVNUM

ERROR ITEM 2  
(MESSAGE)  
(MESSAGE)  
TST NUM ERR PC DRIVE  
STESTN SERRPC DRVNUM  
RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA  
T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA  
RKBA RKWC  
T.BA T.WC

ERROR ITEM 3  
(MESSAGE)  
TST NUM ERR PC DRIVE  
STESTN SERRPC DRVNUM  
RKCS1 RKCS2 RKDS RKER RKASOF RKMRI  
T.CS1 T.CS2 T.DS T.ER T.ASF T.MRI

ERROR ITEMS 4, 5, 6, AND 7 ARE USED TO REPORT ERRORS THAT ARE THE RESULT OF A HARDWARE ERROR INDICATOR BEING SET WHEN NOT EXPECTED, NOT SET WHEN IT IS EXPECTED, OR BOTH. THE ERROR REPORT WILL CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2) ALL THE EXPECTED ERRORS THAT DID NOT OCCUR, OR (3) ALL THE EXPECTED BUT NOT SET ERRORS AND THE UNEXPECTED BUT SET ERRORS.

THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT. INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING STATEMENTS:

"THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"  
"THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"  
"THE ABOVE ARE ERRORS SET IN OPERATION:"

PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT SPECIFY THE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE THAT IDENTIFIES THE OPERATION BEING PERFORMED.

EXAMPLE:  
NON-EXISTANT DRIVE  
THE ABOVE ARE ERRORS SET IN OPERATION:  
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.

EXAMPLE:

NON-EXISTANT DRIVE

THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:  
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR,  
I.E. A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.

EXAMPLE:

NON-EXISTANT MEMORY

THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:

UNIBUS PARITY ERROR

THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:  
WRITE DATA

(ADDITIONAL LINES OF INFORMATION)

THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN  
UNIBUS PARITY ERROR WAS EXPECTED.

ERROR ITEM 4

(DESCRIPTION OF ERROR)

ERROR IN OPERATION

(DESCRIPTION OF OPERATION)

TST NUM ERR PC DRIVE

STESTN SERRPC DRVNUM

RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA

T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA

RKBA RKWC

T.BA T.WA

A00 B00 A01 B01 A02 B02 A03 B03

\$REG10 \$REG11 \$REG12 \$REG13 \$REG14 \$REG15 \$REG16 \$REG17

THE ERRORS REPORTED BY THIS FORMAT ARE:

CONTROLLER DETECTED DRIVE BUS ERROR

DRIVE DETECTED DRIVE BUS ERROR

SEEK INCOMPLETE

NON-EXECUTABLE DRIVE FUNCTION

DRIVE TIMING ERROR

DRIVE UNSAFE

AC LOW

SPINDLE SPEED LOSS

DRIVE OFF TRACK

ILLEGAL DRIVE ADDRESS ERROR

CYLINDER OVERFLOW

DRIVE TYPE ERROR

FORMAT ERROR

WRITE LOCK ERROR

ERROR ITEM 5  
THIS ENTRY IS THE SAME AS ITEM 4 WITH THE ADDITION OF A  
MESSAGE THAT FOLLOWS. THIS MESSAGE IS:

"ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"

THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR ADD  
THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.

ERROR ITEM 6

(DESCRIPTION OF ERROR)  
ERROR IN OPERATION  
(DESCRIPTION OF OPERATION)  
TST NUM ERR PC DRIVE  
STESTN SERRPC DRVNUM  
RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA  
T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA  
RKBA RKWC  
T.BA T.WC

THE ERRORS REPORTED BY THIS FORMAT ARE:

DATA CHECK  
WRITE CHECK  
ECC HARD  
DATA LATE  
OPERATION INCOMPLETE  
HEADER VRC ERROR  
BAD SECTOR ERROR

ERROR ITEM 7

(DESCRIPTION OF ERROR)  
ERROR IN OPERATION  
(DESCRIPTION OF OPERATION)  
TST NUM ERR PC DRIVE  
STESTN SERRPC DRVNUM  
RKCS1 RKCS2 RKDS RKER RKASOF  
T.CS1 T.CS2 T.DS T.ER T.ASOF

THE ERRORS REPORTED BY THIS FORMAT ARE:

NON-EXISTANT DRIVE  
NON-EXISTANT MEMORY  
CONTROLLER TIME OUT  
UNIT FIELD ERROR  
MULTIPLE DRIVE SELECT  
PROGRAMMING ERROR  
UNIBUS PARITY ERROR  
ILLEGAL FUNCTION CODE

DESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR  
ILLEGAL.

ERROR ITEM 10

(DESCRIPTION OF ERROR)  
ERROR AT COMPLETION OF OPERATION



(DESCRIPTION OF OPERATION)  
 TST NUM ERR PC DRIVE  
 \$TESTN \$ERRPC \$DRVNUM  
 EXPT IS  
 \$REG10 \$REG11

THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY  
 COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS. THE SPECIFIC  
 ERRORS ARE:

WORD COUNT INCORRECT  
 BUS ADDRESS INCORRECT  
 CYLINDER ADDRESS INCORRECT  
 TRACK ADDRESS INCORRECT  
 SECTOR ADDRESS INCORRECT

ERROR ITEM 11

(ERROR INDICATOR OR STATUS BIT)  
 NOT SET AS A RESULT OF  
 (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION  
 TST NUM ERR PC DRIVE  
 \$TESTN \$ERRPC \$DRVNUM  
 RKCS1 RKCS2 RKDS RKER RKASOF RKMRI  
 T.CS1 T.CS2 T.DS T.ER T.ASOF T.MRI

ERROR ITEM 12

THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

"NOT RESET AS A RESULT OF"

ERROR ITEM 13

THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

"SET AS A RESULT OF"

ERROR ITEM 14

THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:

"RESET AS A RESULT OF"

ERROR ITEM 15

(HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)  
 TST NUM ERR PC DRIVE  
 \$TESTN \$ERRPC \$DRVNUM  
 GOOD BAD WORD NUM  
 \$REG10 \$REG11 \$REG12

ERROR ITEM 16

ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15.

6.0

SUBROUTINES

IN THE INTEREST OF CONSERVING MEMORY, IT IS NECESSARY TO MAKE  
 EXTENSIVE USE OF SUBROUTINES. HOWEVER, IN THE INTEREST OF  
 PRESERVING CODE READABILITY, SUBROUTINE NAMING IS DESCRIPTIVE  
 OF THE FUNCTION PERFORMED. THE SUBROUTINE FUNCTION IS KEPT

SMALL AND IN GENERAL A SUBROUTINE ONLY PERFORMS ONE FUNCTION, I.E., LOAD THE RK611 REGISTER AND START AN OPERATION (TLOADRK) OR WAIT A SPECIFIED NUMBER OF MILLISECONDS FOR AN INTERRUPT (TWATNN WHERE NN VARIES FROM CALL TO CALL AND IS THE TIME TO WAIT). THE FOLLOWING IS A DESCRIPTION OF THE SUBROUTINES NOT PROVIDED BY SYSMAC:

\*\*\*\*\*  
 LINE CLOCK CALIBRATE  
 \*\*\*\*\*

WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE LINE CLOCK SIMULATOR.

CALL: JSR PC,CLKCAL

\*\*\*\*\*  
 OPTION PRESENT TEST AND SETUP  
 \*\*\*\*\*

THIS ROUTINE CHECKS IF THE MEMORU PARITY OPTION AND THE LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.

CALL: JSR PC,OPTTST

\*\*\*\*\*  
 LINE CLOCK SIMULATION ROUTINE  
 \*\*\*\*\*

THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO DO THIS THE VALUE STORED IN MILCNT IS USED AS THE BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK TICK).

THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.

CALL: JSR PC,MYTIME

\*\*\*\*\*  
 WAIT FOR INTERRUPT ROUTINE  
 \*\*\*\*\*

THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE WHILE WAITING FOR INTERRUPT. IF

INTERRUPT DOES NOT OCCUR IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED (BUT NOT PRINTED IN THE ROUTINE) AND THEN RETURNS TO THE LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.

CALL: TWAT16 THROUGH TWAT159, TWAT1S, TWAT2S,  
TWATBS, AND TWATIM

\*\*\*\*\*  
"L" REGISTER LOADING ROUTINE  
\*\*\*\*\*

THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.

CALL: JSR R4,LRLoad  
COMMAND  
WORD COUNT  
BUS ADDRESS  
.BYTE SECTOR ADDRESS  
.BYTE TRACK ADDRESS  
CYLINDER ADDRESS

\*\*\*\*\*  
LOAD RK611 FOR OPERATION  
\*\*\*\*\*

THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER. THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION OF THE REGISTER CONTENTS. L.CS1 IS TRANSFERRED LAST AS IT SHOULD BE IF THE GO BIT IS SET.

CALL: TLOADRK

\*\*\*\*\*  
STORE RK611 REGISTERS  
\*\*\*\*\*

ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN THIS ROUTINE.

CALL: TGETRK

\*\*\*\*\*  
BIT COUNTER IN A WORD  
\*\*\*\*\*

THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK BY THE CALLING ROUTINE. THE NUMBER OF BITS

FOUND IN THE WORD ARE PASSED BACK ON THE STACK.

CALL: JSR R4,BITCNT

\*\*\*\*\*  
 MAINTENANCE CLOCK ROUTINE  
 \*\*\*\*\*

THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE ADDRESS AFTER THE CALL. THE FIRST BYTE CONTAINS THE NUMBER OF PHASE ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF CLOCK TRANSITIONS (PARTIAL PHASES) TO BE DONE.

CALL: JSR R4,MCLOCK  
       .BYTE               ;NUMBER OF CLOCK TRANSITIONS  
       .BYTE               ;NUMBER OF PHASE ADDRESSES

\*\*\*\*\*  
 READ AND SORT HEADERS  
 \*\*\*\*\*

THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN ASCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER EACH READ OF THE DATA BUFFER.

CALL: JSR R4,RDSTHD  
       TCHKOP             ;RETURN POINT IF CERR IN READ  
                          ;HDR  
       ERROR 4            ;OR 5, 6, 7  
       ERROR 13          ;RETURN IF DATA LATE IN DB  
                          ;UNLOAD  
       ERROR 2            ;RETURN IF TOO SLOW OR  
                          ;IF HDR 0 NOT FOUND

\*\*\*\*\*  
 GET DRIVE STATUS  
 \*\*\*\*\*

THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN \$REG10 THROUGH \$REG17. THESE REGISTERS ARE FIRST CLEARED TO ALL ONES AND THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT IN THE REGISTERS.

CALL: JSR R4,GETDRS  
       BR                ERROR PROCESSING        ERROR RETURN  
       BR                NO ERROR PROCESSING     GOOD RETURN

\*\*\*\*\*  
 SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST  
 \*\*\*\*\*

THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR COMMAND. CERR AND DI ARE MONITORED FOR A SHORT PERIOD OF TIME DURING WHICH THEY SHOULD BOTH RESET.

IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.

IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN SKIPS OVER THE ERROR CALL AND TEST ABORT.

THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY AN ERROR MESSAGE AND BRANCH TO END OF TEST. THIS IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO THE TEST.

CALL: TSSINIT

\*\*\*\*\*  
WORD COUNT AT END OF OPERATION CHECK  
\*\*\*\*\*

THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF UNEQUAL, THE ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)

CALL: JSR R4,CHKWC  
.WORD ;EXPECTED WC VALUE

\*\*\*\*\*  
BUS ADDRESS AT END OF OPERATION CHECK  
\*\*\*\*\*

THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT. IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)

IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS ADDRESS IS THE STARTING BUS ADDRESS.

CALL: JSR R4,CHKBA

\*\*\*\*\*  
CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION  
\*\*\*\*\*

THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE OPERATION. THE CONTENTS OF THE LOAD REGISTER STORAGE ARE COUNTED ON TO HAVE THE INITIAL

VALUES TO MAKE THE NECESSARY CALCULATION.

ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES STORAGE EXPCYL, EXPTRK, EXPSEC. ALL 3 ARE CHECKED AND IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE ERROR FLAGS FIELD (GRP4ER) IS SET.

CALL: JSR R4,CHKCTS

\*\*\*\*\*  
OPERATION CHECK ROUTINE  
\*\*\*\*\*

THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME SOFTWARE ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS: THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL BE THE PROPER ITEM FOR THE FORMAT REQUIRED BY THIS ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4, 5, 6, 7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT FORMAT.

FOR NO EXPECTED ERRORS:  
CALL: TCHKOP

FOR EXPECTED ERRORS:  
CALL: TCHKWE

.WORD ;GROUP 1 EXPECTED ERRORS  
.WORD ;GROUP 2 EXPECTED ERRORS  
.WORD ;GROUP 3 EXPECTED ERRORS

WHERE EACH BIT IN THE THREE WORDS FOLLOWING THE CALL REPRESENT A SPECIFIC ERROR. THE BIT ASSIGNMENTS ARE GIVEN BELOW:

GROUP 1 ERRORS:  
BIT 0 - CONTROLLER DETECTED DRIVE BUS  
PARITY ERROR  
BIT 1 - SEEK INCOMPLETE  
BIT 2 - NON-EXECUTABLE DRIVE FUNCTION  
BIT 3 - DRIVE DETECTED DRIVE BUS PARITY ERROR  
BIT 4 - FORMAT ERROR  
BIT 5 - DRIVE TYPE ERROR  
BIT 6 - AC LOW ERROR  
BIT 7 - SPEED LOSS ERROR  
BIT 8 - DRIVE OFF TRACK ERROR  
BIT 9 - CYLINDER OVERFLOW ERROR  
BIT 10 - ILLEGAL DISK ADDRESS ERROR  
BIT 11 - WRITE LOCK ERROR  
BIT 12 - DRIVE TIMING ERROR

BIT 13 - NO CERR WITH OTHER ERROR SET ERROR  
 BIT 14 - DRIVE UNSAFE ERROR  
 BIT 15 - CERR BUT NO OTHER ERROR SET ERROR

GROUP 2 ERRORS:  
 BIT 0 - ECC HARD ERROR  
 BIT 1 - DATA CHECK ERROR  
 BIT 2 - WRITE CHECK ERROR  
 BIT 3 - DATA LATE ERROR  
 BIT 4 - OPERATION INCOMPLETE ERROR  
 BIT 5 - HEADER VRC ERROR  
 BIT 6 - BAD SECTOR ERROR

GROUP 2 ERRORS:  
 BIT 0 - NON-EXISTANT DRIVE ERROR  
 BIT 1 - CONTROLLER TIMEOUT ERROR  
 BIT 2 - UNIT FIELD ERROR  
 BIT 3 - MULTIPLE DRIVE SELECT ERROR  
 BIT 4 - PROGRAMMING ERROR  
 BIT 5 - NON-EXISTANT MEMORY ERROR  
 BIT 6 - UNIBUS PARITY ERROR  
 BIT 7 - ILLEGAL FUNTION ERROR

\*\*\*\*\*  
 BAD SECTOR CHECK  
 \*\*\*\*\*

THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS IS PLACED ON THE STACK AFTER THE ENTIRE FIELD IS SEARCHED.

CALL: JSR R4,BDSRCK  
 <ADDRESS OF FIELD TO BE SEARCHED>

\*\*\*\*\*  
 DATA GENERATION AND COMPARE ROUTINE  
 \*\*\*\*\*

CALLS: JSR R4,GENCOM  
 CONTROL WORD  
  
 JSR R4,GENCOM  
 CONTROL WORD  
 LENGTH  
  
 JSR R4,GENCOM  
 CONTROL WORD  
 RELOCATION CONSTANT  
 LENGTH

RETURN:        RTS     R4

R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:  
THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE  
CONTROL WORD. THIS IS UNCONDITIONAL.

THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE  
LENGTH IF THE OPERATION REQUIRES DATA COMPARE AND DATA  
MISCOMPARED. IF DATA IS TO BE GENERATED ONLY OR NO  
DATA COMPARE ERRORS OCCURRED, THE RETURN IS TO LENGTH  
+4.

THE THIRD CALL IS IDENTICAL TO THE SECOND.

DEFINITION OF CONTROL WORD:

- BIT 15     - DO COMPARE OPERATION OF IBUFF (SOURCE) TO  
           - OBUFF (DESTINATION). EXPECTED VALUES ARE  
           - IN OBUFF (DESTINATION).
- BIT 14     - RESUME COMPARE OPERATION FROM POINT LEFT  
           - BY LAST COMPARE.
- BIT 13     - INVOKE MEMORY MANAGEMENT FOR SOURCE  
           - (IBUFF).
- BIT 12     - INVOKE MEMORY MANAGEMENT FOR DESTINATION  
           - (OBUFF).
- BIT 11     - REPEAT FIRST WORD OF SELECTED PATTERN  
           - THROUGHOUT OBUFF.
- BIT 10     - CLEAR IBUFF TO PATTERN SELECTED.
- BIT 9      - BUILD HEADERS, CONSIDERING BS FILES
- BIT 8      - BUILD HEADERS, ALL SECTORS INDICATE GOOD  
           - SECTORS.
- BIT 7      - HEADER OPERATION SPECIFIED (EITHER  
           - COMPARE OR BUILD).
- BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0  
           - INDICATES NO DATA GENERATION, 1 IS ALL  
           - ZEROS, AND 7 IS ALL ONES. OTHER PATTERNS  
           - PROVIDED ARE PATTERNS 2-6, 8-16. REFER  
           - TO THE PROGRAM LISTING FOR PAT02 THROUGH  
           - PAT16.

EXPLANATION OF CALLS:

THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS  
USED FOR BUILDING OR COMPARING HEADERS OR RESUMING  
A COMPARE OPERATION.

THE CALL WITH CONTROL WORD AND LENGTH AS  
PARAMETERS IS USED FOR DATA GENERATION OR COMPARE  
AND FOR IBUFF INITIALIZATION.

THE CALL WITH CONTROL WORD, RELOCATION CONSTANT,  
AND LENGTH IS USED FOR DATA GENERATION OR COMPARE  
WITH MEMORY MANAGEMENT.

DESCRIPTION:



THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:

- A. BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8) OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK ARE LISTED THEREIN (BIT 9).
- B. COMPARE THE CONTENTS OF IBUFF AND OBUFF (BIT 15). THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8) WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS MAY BE BUILT BEFORE THE COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
- C. RESUME COMPARE OPERATION. IF A COMPARE OPERATION DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14 SET IN THE CONTROL WORD.
- D. DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT. MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO ANY AREA OF AVAILABLE MEMORY.

\*\*\*\*\*  
 PHASE LOCKED LOOP CLOCK ADJUSTMENT ROUTINE  
 \*\*\*\*\*

THIS ROUTINE IS ENTERED VIA START LOCATION 220(8). THE PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE. THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND RESETS THE DIAGNOSTIC MODE BIT IN RKMRI. INSTRUCTIONS ON WHERE TO SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.

THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.

21	OPERATIONAL SWITCH SETTINGS
33	BASIC DEFINITIONS
143	MEMORY MANAGEMENT DEFINITIONS
182	RK611 CONTROLLER REGISTER DEFINITION
201	DRIVE COMMANDS
218	CONTROL AND STATUS REGISTER 1 BITS
234	CONTROL AND STATUS REGISTER 2 BITS
251	ERROR REGISTER BIT DEFINITION
270	STATUS REGISTER BIT DEFINITION
286	MAINTENANCE REGISTER 1 BIT DEFINITION
303	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
316	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
329	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
343	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
357	COMMON MASKS
367	TRAP CATCHER
392	STARTING ADDRES(ES)
399	APT PARAMETER BLOCK
421	COMMON TAGS
499	APT MAILBOX-ETABLE
548	ERROR POINTER TABLE
853	REGISTER STORAGE FOR TEST
872	REGISTER SETUP STORAGE
883	PROGRAM DEFINED VARIABLES
939	PROGRAM SETUP
958	INITIALIZE THE COMMON TAGS
1001	TYPE PROGRAM NAME
1008	GET VALUE FOR SOFTWARE SWITCH REGISTER
1108	**BASIC INTERFACE AND OPTION TESTS
1109	T1 RK611 BASE ADDRESS TEST
1140	T2 INTERRUPT VECTOR ADDRESS TEST
1167	**STATUS VALID TESTS
1170	T3 SELECT ALL DRIVES
1411	T4 RELEASE ALL DRIVES
1449	T5 NON-STANDARD MESSAGES AND SVAL
1502	T6 WRITING CS2 AND STATUS VALID
1545	**CONTROLLER ERROR TESTS
1547	T7 DRIVE TYPE ERROR
1581	T10 STATUS VALID AND PARITY ERROR
1768	T11 UNIT FIELD ERROR ON RELEASE
1816	T12 UNIT FIELD ERROR ON SELECT
1863	**ATTENTION HANDLING BY CONTROLLER
1865	T13 DOUBLE INTERRUPT
1937	T14 SINGLE INTERRUPT FROM ATTENTION
1981	T15 RESET ATTENTIONS WITH UNIBUS INIT
2037	**ILLEGAL DISK ADDRESS ERROR TESTS
2039	T16 ILLEGAL DISK ADDRESS (PART 1)
2106	T17 ILLEGAL DISK ADDRESS (PART 2)
2151	**WRITE HEADER TESTS
2153	T20 READ BAD SECTOR INFORMATION
2311	T21 FORMAT IN 26 SECTOR FORMAT
2429	**HEADER RECOGNITION TESTS
2431	T22 BAD SECTOR ERROR
2513	T23 HEADER VRC ERROR
2593	T24 BAD SECTOR ERROR AND HVRC ERROR
2646	T25 OPERATION INCOMPLETE

2698	T26	OPI WITH HVRC ERROR
2759	T27	HVRC IGNORE ON NON-ADDRESSED SECTOR
2816	**DATA	TRANSFER TESTS
2818	T30	WRITE AND READ ONE SECTOR
2914	T31	WRITE DATA ADDRESS GREATER THAN 32K
2996	T32	READ DATA ADDRESS GREATER THAN 32K
3065	T33	WRITE DATA ADDRESS GREATER THAN 64K
3145	T34	READ DATA ADDRESS GREATER THAN 64K
3216	T35	WRITE DATA ADDRESS GREATER THAN 96K
3298	T36	READ DATA ADDRESS GREATER THAN 96K
3370	T37	PARTIAL SECTOR WRITE DATA
3444	T40	PARTIAL SECTOR READ DATA
3517	T41	WRITE DATA WITH NON-EXISTENT MEMORY
3546	T42	READ DATA WITH NON-EXISTENT MEMORY
3575	T43	UNIBUS PARITY ERROR
3662	**MULTIPLE	SECTOR OPERATIONS
3664	T44	TWO SECTOR WRITE DATA (PART 1)
3748	T45	TWO SECTOR WRITE DATA (PART 2)
3836	T46	TWO SECTOR WRITE DATA (PART 3)
3921	T47	MID-TRANSFER SEEK ON WRITE (PART 1)
4007	T50	MID-TRANSFER SEEK ON WRITE (PART 2)
4093	T51	TWO SECTOR READ DATA (PART 1)
4153	T52	TWO SECTOR READ DATA (PART 2)
4211	T53	TWO SECTOR READ DATA (PART 3)
4269	T54	MID-TRANSFER SEEK ON READ (PART 1)
4326	T55	MID-TRANSFER SEEK ON READ (PART 2)
4383	T56	CYLINDER ADDRESS OVERFLOW (PART 1)
4411	T57	CYLINDER ADDRESS OVERFLOW (PART 2)
4444	**18 BIT	DATA TRANSFER TESTS
4446	T60	FORMAT IN 24 SECTOR FORMAT
4526	T61	24 SECTOR FORMAT DATA TRANSFER (PART 1)
4595	T62	24 SECTOR FORMAT DATA TRANSFER (PART 2)
4667	T63	24 SECTOR FORMAT DATA TRANSFER (PART 3)
4739	**SPECIAL	DATA TRANSFER TESTS
4741	T64	MULTI-SECTOR DATA TRANSFER AND BSE
4901	T65	FORMAT TEST
4975	**WRITE	CHECK TESTS
4977	T66	WRITE-CHECK WITH NO ERROR
5024	T67	WRITE CHECK ERROR (PART 1)
5158	T70	WRITE CHECK ERROR (PART 2)
5294	T71	WRITE CHECK OF PARTIAL SECTOR
5346	**MAXIMUM	DATA TRANSFER AND CONTROLLER TIME OUT
5348	T72	MAXIMUM DATA TRANSFER (PART 1)
5554	T73	MAXIMUM DATA TRANSFER (PART 2)
5729	T74	CONTROLLER TIME OUT
5818	**ERRORS	DURING DATA TRANSFER
5821	T75	LIMIT DETECT ON DATA TRANSFER
5925	T76	PROGRAMMING ERROR
5961	T77	ECC HARD
6024	T100	DRIVE TIMING ERROR
6114	**ERROR	FORCING IN DRIVE
6116	T101	INITIALIZE CLEARING SACK
6173	T102	DRIVE OFF TRACK
6324	T103	FILE UNSAFE
6533	T104	DUMMY TEST FOR PREVIOUS TEST EXIT
6560	**MULTI-	DRIVE OPERATIONS

6579	T105	RESET ATTENTIONS WITH UNIBUS INIT
6634	T106	RESET ATTENTIONS WITH SUBSYSTEM CLEAR
6688	T107	SVAL AND ATTENTION FROM OTHER DRIVE
6853	T110	OVERLAPPED OPERATIONS
7034		END OF PASS ROUTINE
7072		ROUTINE TO SIZE MEMORY
7149		SCOPE HANDLER ROUTINE
7304		APT COMMUNICATIONS ROUTINE
7361		ERROR HANDLER ROUTINE
7439		TYPE ERROR ROUTINE
7516		NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
7524		RK611 INTERRUPT HANDLER
7533		MEMORY PARITY ERROR TRAP HANDLER
7545		LINE CLOCK INTERRUPT HANDLER
7554		OPTION PRESENT TEST AND SETUP
7604		LOOP ON INTERNAL ERROR
7615		LINE CLOCK CALIBRATE
7633		LINE CLOCK SIMULATION ROUTINE
7660		WAIT FOR INTERRUPT ROUTINE
7738		"L" REGISTER LOADING ROUTINE
7767		LOAD RK611 FOR OPERATION
7792		STORE RK611 REGISTERS
7836		MAINTENANCE CLOCK ROUTINE
7859		READ AND SORT HEADERS
7957		GET DRIVE STATUS
8010		SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
8065		WORD COUNT AT END OF OPERATION CHECK
8081		BUS ADDRESS AT END OF OPERATION CHECK
8140		CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION
8221		OPERATION CHECK ROUTINE
8682		BAD SECTOR CHECK
8723		DATA GENERATION AND COMPARE ROUTINE
9111		PHASE LOCK LOOP CLOCK ADJUSTMENT ROUTINE
9150		TYPE ROUTINE
9229		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9296		BINARY TO OCTAL (ASCII) AND TYPE
9373		TTY INPUT ROUTINE
9630		READ AN OCTAL NUMBER FROM THE TTY
9683		SAVE AND RESTORE R0-R5 ROUTINES
9728		POWER DOWN AND UP ROUTINES
9771		TRAP DECODER
9794		TRAP TABLE
9837		DATA PATTERNS
10084		FIELDS AND VARIABLES FOR OPERATION CHECKING
10202		TABLE OF OPERATION MESSAGE ADDRESS
10223		OPERATION MESSAGES
10279		ASCII MESSAGES
10500	ERROR MESSAGES	
10870		DATA HEADERS FOR ERROR REPORTS
11004		DATA TABLES FOR ERROR REPORTS
11025		DATA FORMATS FOR ERROR REPORTS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

167400  
000001

001100

000011  
000012  
000015  
000200  
177776

177774  
177772  
177570  
177570

000000  
000001  
000002  
000003  
000004

```

.NLIST  CND,MD,MC
.LIST   ME
.ENABL  ABS,AMA
.MCALL  .HEADER,.SWRHI,.SWRLO,.EQUAT,.SETUP,$4OCAT,$CMTAG
.MCALL  .SEOP,$SCOPE,$ERROR,$ERRTYP,$TYPE,$TYPOCT
.MCALL  .$POWER,$STRAP,$READ,$RDOCT,$KT1,$SIZE,$ACT11
.MCALL  .$APTYPE,$SAVE,TYPNAM,$STYPDEC,$APTHDR
$SWR=   167400
$STN=   1
.TITLE  RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
;*COPYRIGHT (C) 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY MARV TEGROTENHUIS
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
;*
.SBTTL  OPERATIONAL SWITCH SETTINGS
;*
;*      SWITCH          USE
;*      -----          -
;*      15              HALT ON ERROR
;*      14              LOOP ON TEST
;*      13              INHIBIT ERROR TYPEOUTS
;*      12              ABORT PROGRAM AFTER 20 ERRORS
;*      11              INHIBIT ITERATIONS
;*      10              BELL ON ERROR
;*      9               LOOP ON ERROR
;*      8               LOOP ON TEST IN SWR<7:0>
.SBTTL  BASIC DEFINITIONS
;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK=  1100
.EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV  IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
;*MISCELLANEOUS DEFINITIONS
HT=     11              ;;CODE FOR HORIZONTAL TAB
LF=     12              ;;CODE FOR LINE FEED
CR=     15              ;;CODE FOR CARRIAGE RETURN
CRLF=   200            ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS=     177776         ;;PROCESSOR STATUS WORD
.EQUIV  PS,PSW
STKLMT= 177774         ;;STACK LIMIT REGISTER
PIRQ=   177772         ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR=   177570         ;;HARDWARE SWITCH REGISTER
DDISP=  177570         ;;HARDWARE DISPLAY REGISTER
;*GENERAL PURPOSE REGISTER DEFINITIONS
R0=     %0             ;;GENERAL REGISTER
R1=     %1             ;;GENERAL REGISTER
R2=     %2             ;;GENERAL REGISTER
R3=     %3             ;;GENERAL REGISTER
R4=     %4             ;;GENERAL REGISTER

```

DZR6KB.P11

## BASIC DEFINITIONS

```

57      000005      R5=      %5          ;; GENERAL REGISTER
58      000006      R6=      %6          ;; GENERAL REGISTER
59      000007      R7=      %7          ;; GENERAL REGISTER
60      .EQUIV      R6,SP      ;; STACK POINTER
61      .EQUIV      R7,PC      ;; PROGRAM COUNTER
62
63      ;*PRIORITY LEVEL DEFINITIONS
64      000000      PR0=      0          ;; PRIORITY LEVEL 0
65      000040      PR1=      40         ;; PRIORITY LEVEL 1
66      000100      PR2=      100        ;; PRIORITY LEVEL 2
67      000140      PR3=      140        ;; PRIORITY LEVEL 3
68      000200      PR4=      200        ;; PRIORITY LEVEL 4
69      000240      PR5=      240        ;; PRIORITY LEVEL 5
70      000300      PR6=      300        ;; PRIORITY LEVEL 6
71      000340      PR7=      340        ;; PRIORITY LEVEL 7
72
73      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
74      100000      SW15=     100000
75      040000      SW14=     40000
76      020000      SW13=     20000
77      010000      SW12=     10000
78      004000      SW11=     4000
79      002000      SW10=     2000
80      001000      SW09=     1000
81      000400      SW08=     400
82      000200      SW07=     200
83      000100      SW06=     100
84      000040      SW05=     40
85      000020      SW04=     20
86      000010      SW03=     10
87      000004      SW02=     4
88      000002      SW01=     2
89      000001      SW00=     1
90      .EQUIV      SW09,SW9
91      .EQUIV      SW08,SW8
92      .EQUIV      SW07,SW7
93      .EQUIV      SW06,SW6
94      .EQUIV      SW05,SW5
95      .EQUIV      SW04,SW4
96      .EQUIV      SW03,SW3
97      .EQUIV      SW02,SW2
98      .EQUIV      SW01,SW1
99      .EQUIV      SW00,SW0
100
101      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
102      100000      BIT15=    100000
103      040000      BIT14=    40000
104      020000      BIT13=    20000
105      010000      BIT12=    10000
106      004000      BIT11=    4000
107      002000      BIT10=    2000
108      001000      BIT09=    1000
109      000400      BIT08=    400
110      000200      BIT07=    200
111      000100      BIT06=    100
112      000040      BIT05=    40

```

```

113      000020      BIT04= 20
114      000010      BIT03= 10
115      000004      BIT02= 4
116      000002      BIT01= 2
117      000001      BIT00= 1
118      .EQUIV      BIT09,BIT9
119      .EQUIV      BIT08,BIT8
120      .EQUIV      BIT07,BIT7
121      .EQUIV      BIT06,BIT6
122      .EQUIV      BIT05,BIT5
123      .EQUIV      BIT04,BIT4
124      .EQUIV      BIT03,BIT3
125      .EQUIV      BIT02,BIT2
126      .EQUIV      BIT01,BIT1
127      .EQUIV      BIT00,BIT0
128
129      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
130      000004      ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
131      000010      RESVEC= 10         ;; RESERVED AND ILLEGAL INSTRUCTIONS
132      000014      TBITVEC=14         ;; "T" BIT
133      000014      TRTVEC= 14         ;; TRACE TRAP
134      000014      BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
135      000020      IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
136      000024      PWRVEC= 24         ;; POWER FAIL
137      000030      EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
138      000034      TRAPVEC=34         ;; "TRAP" TRAP
139      000060      TKVEC= 60          ;; TTY KEYBOARD VECTOR
140      000064      TPVEC= 64          ;; TTY PRINTER VECTOR
141      000240      PIRQVEC=240        ;; PROGRAM INTERRUPT REQUEST VECTOR
142      .SBTTL      MEMORY MANAGEMENT DEFINITIONS
143
144      ;*KT11 VECTOR ADDRESS
145
146      000250      MMVEC= 250
147
148      ;*KT11 STATUS REGISTER ADDRESSES
149
150      177572      SR0= 177572
151      177574      SR1= 177574
152      177576      SR2= 177576
153      172516      SR3= 172516
154
155      ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
156
157      172300      KIPDR0= 172300
158      172302      KIPDR1= 172302
159      172304      KIPDR2= 172304
160      172306      KIPDR3= 172306
161      172310      KIPDR4= 172310
162      172312      KIPDR5= 172312
163      172314      KIPDR6= 172314
164      172316      KIPDR7= 172316
165
166      ;*KERNEL "I" PAGE ADDRESS REGISTERS
167
168      172340      KIPAR0= 172340

```

```

169      172342      KIPAR1= 172342
170      172344      KIPAR2= 172344
171      172346      KIPAR3= 172346
172      172350      KIPAR4= 172350
173      172352      KIPAR5= 172352
174      172354      KIPAR6= 172354
175      172356      KIPAR7= 172356
176
177      000210      AVECT1= 210      ;DEFINE RK611 VECTOR INTERRUPT
178      000240      APRIOR= PR5      ;DEFINE RK611 PRIORITY
179      177440      ABASE= 177440    ;DEFINE RK611 BASE BUS ADDRESS
180
181      .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
182
183      000000      RKCS1= 0      ;CONTROL AND STATUS REGISTER 1
184      000002      RKWC= 2      ;WORD COUNT REGISTER
185      000004      RKBA= 4      ;BUS ADDRESS REGISTER
186      000006      RKDA= 6      ;DESIRED TRACK SECTOR REGISTER
187      000010      RKCS2= 10     ;CONTROL AND STATUS REGISTER 2
188      000012      RKDS= 12     ;DRIVE STATUS REGISTER
189      000014      RKER= 14     ;ERROR REGISTER
190      000016      RKASOF= 16    ;ATTENTION SUMMARY AND OFFSET REGISTER
191      000020      RKDCYL= 20    ;DESIRED CYLINDER REGISTER
192      000024      RKDB= 24     ;DATA BUFFER
193      000026      RKMR1= 26    ;MAINTENANCE REGISTER 1
194      000034      RKMR2= 34    ;MAINTENANCE REGISTER 2
195      000036      RKMR3= 36    ;MAINTENANCE REGISTER 3
196      000030      RKECPS= 30    ;ECC POSITION INFORMATION
197      000032      RKECPT= 32    ;ECC PATTERN INFORMATION
198      000022      RKSPAR= 22    ;SPARE REGISTER
199
200      .SBTTL  DRIVE COMMANDS
201
202      000101      SELDRV= 101    ;SELECT DRIVE
203      000103      PACK= 103    ;PACK ACKNOWLEDGE
204      000105      CLEAR= 105    ;DRIVE CLEAR
205      000107      UNLOAD= 107   ;UNLOAD
206      000111      SRTSPL= 111   ;START SPINDLE
207      000113      RECAL= 113   ;RECALIBRATE
208      000115      OFFSET= 115  ;OFFSET
209      000117      SEEK= 117    ;SEEK
210      000121      RDDATA= 121   ;READ DATA
211      000123      WRDATA= 123   ;WRITE DATA
212      000125      RDHEAD= 125   ;READ HEADER
213      000127      WRHEAD= 127   ;WRITE HEADER AND DATA
214      000131      WRTCHK= 131   ;WRITE CHECK
215      000300      INTR= 300    ;GENERATE INTERRUPT TO CPU
216
217      .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
218
219      000001      GO= BIT0      ;GO BIT
220      000100      IE= BIT6      ;INTERRUPT ENABLE
221      000200      RDY= BIT7      ;CONTROLLER READY
222      000400      BA16= BIT8     ;BUS ADDRESS BIT 16
223      001000      BA17= BIT9     ;BUS ADDRESS BIT 17
224      002000      CDT= BIT10    ;CONTROLLER DRIVE TYPE (0=RK06)

```



225	004000	CTO=	BIT11	; CONTROLLER TIMED OUT WAITING FOR
226				; DRIVE RESPONSE
227	010000	CFMT=	BIT12	; CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
228	020000	SPAR=	BIT13	; DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
229	040000	DI=	BIT14	; DRIVE INTERRUPT
230	100000	CERR=	BIT15	; CONTROLLER ERROR
231	100000	CCLR=	BIT15	; CONTROLLER CLEAR

.SBTTL CONTROL AND STATUS REGISTER 2 BITS

235	000007	DRVMSK=	7	; MASK FOR DRIVE SELECTION CODE
236	000010	RLS=	BIT3	; DESELECT OR RELEASE DRIVE IN BITS 0-2
237	000020	BAI=	BIT4	; BUS ADDRESS INCREMENT INHIBIT
238	000040	SCLR=	BIT5	; CLEAR CONTROLLER AND ALL DRIVES
239	000100	IR=	BIT6	; INPUT READY
240	000200	OR=	BIT7	; OUTPUT READY
241	000400	UFE=	BIT8	; UNIT FIELD ERROR
242	001000	MDS=	BIT9	; MULTIPLE DRIVE SELECT
243	002000	PGE=	BIT10	; PROGRAMMING ERROR
244	004000	NEM=	BIT11	; NON-EXISTENT MEMORY
245	010000	NED=	BIT12	; NON-EXISTENT DRIVE
246	020000	UPE=	BIT13	; UNIBUS PARITY ERROR
247	040000	WCE=	BIT14	; WRITE CHECK ERROR
248	100000	DLT=	BIT15	; DATA LATE ERROR

.SBTTL ERROR REGISTER BIT DEFINITION

252	000001	ILF=	BIT0	; ILLEGAL FUNCTION CODE
253	000002	SKI=	BIT1	; SEEK INCOMPLETE
254	000004	NXF=	BIT2	; NON-EXECUTABLE DRIVE FUNCTION
255	000010	DRPAR=	BIT3	; DRIVE DETECTED DRIVE BUS PARITY ERROR
256	000020	FMTE=	BIT4	; FORMAT ERROR
257	000040	DTYPE=	BIT5	; DRIVE TYPE ERROR
258	000100	ECH=	BIT6	; ECC HARD
259	000200	BSE=	BIT7	; BAD SECTOR ERROR
260	000400	HVRC=	BIT8	; HEADER VRC ERROR
261	001000	COE=	BIT9	; CYLINDER ADDRESS OVERFLOW ERROR
262	002000	IDAE=	BIT10	; INVALID DISK ADDRESS ERROR
263	004000	WLE=	BIT11	; WRITE LOCK ERROR
264	010000	DTE=	BIT12	; DRIVE TIMING ERROR
265	020000	OPI=	BIT13	; OPERATION (SEARCH) INCOMPLETE
266	040000	UNS=	BIT14	; DRIVE UNSAFE
267	100000	DCK=	BIT15	; DATA CHECK

.SBTTL STATUS REGISTER BIT DEFINITION

271	000001	DRA=	BIT0	; DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
273	000004	OFST=	BIT2	; DRIVE OFFSET
274	000010	ACLO=	BIT3	; AC LOW
275	000020	SPDLSS=	BIT4	; SPEED LOSS
276	000040	DROT=	BIT5	; DRIVE OFF TRACK
277	000100	VV=	BIT6	; VOLUME VALID
278	000200	DRDY=	BIT7	; DRIVE READY
279	000400	DDT=	BIT8	; DRIVE TYPE (0=RK06)
280	004000	WRL=	BIT11	; WRITE LOCK

```

281 020000 PIP= BIT13 ; POSITIONING IN PROGRESS
282 040000 DSC= BIT14 ; DRIVE STATUS CHANGE
283 100000 SVAL= BIT15 ; STATUS VALID
284
285 .SBTTL MAINTENANCE REGISTER 1 BIT DEFINITION
286
287 000017 MESMSK= 17 ; MESSAGE MASK
288
289 000020 PAT= BIT4 ; FORCE EVEN PARITY ON SERCON MESSAGE LINES
290 000040 DMD= BIT5 ; DIAGNOSTIC MODE
291 000100 MSP= BIT6 ; MAINTENANCE SECTOR PULSE
292 000200 MIND= BIT7 ; MAINTENANCE INDEX
293 000400 MCLK= BIT8 ; MAINTENANCE CLOCK
294 001000 MERD= BIT9 ; MAINTENANCE ENCODED READ DATA
295 002000 MEWD= BIT10 ; MAINTENANCE ENCODED WRITE DATA
296 004000 PCA= BIT11 ; PRECOMPENSATION ADVANCE
297 010000 PCD= BIT12 ; PRECOMPENSATION DELAY
298 020000 ECCW= BIT13 ; ECC WORD IS BEING READ OR WRITTEN
299 040000 WRTGAT= BIT14 ; WRITE GATE
300 100000 RDGATE= BIT15 ; READ GATE
301
302 .SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
303
304 000040 S.DRA= BIT5 ; DRIVE AVAILIABLE
305 000100 S.VV= BIT6 ; VOLUME VALID
306 000200 S.DRY= BIT7 ; DRIVE READY
307 000400 S.TYPE= BIT8 ; DRIVE TYPE
308 001000 S.FORM= BIT9 ; DRIVE FORMAT
309 002000 S.OFF= BIT10 ; OFFSET
310 004000 S.WRL= BIT11 ; WRITE LOCK
311 010000 S.SPIN= BIT12 ; SPINDLE ON
312 020000 S.PIP= BIT13 ; POSITIONING IN PROGRESS
313 040000 S.DSC= BIT14 ; DRIVE STATUS CHANGE
314
315 .SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
316
317 000040 S.ICYL= BIT5 ; ILLEGAL CYLINDER ADDRESS
318 000100 S.ACLO= BIT6 ; AC LOW
319 000200 S.FLT= BIT7 ; DRIVE FAULT
320 000400 S.ILF= BIT8 ; ILLEGAL FUNCTION
321 001000 S.PAR= BIT9 ; DRIVE DETECTED SERCON PARITY ERROR
322 002000 S.SKI= BIT10 ; SEEK INCOMPLETE
323 004000 S.WLE= BIT11 ; WRITE LOCK ERROR
324 010000 S.SPLS= BIT12 ; SPEED LOSS
325 020000 S.DROT= BIT13 ; DRIVE OFF TRACK
326 040000 S.UNS= BIT14 ; DRIVE UNSAFE
327
328 .SBTTL DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
329
330 000020 S.XDOK= BIT4 ; TRANSDUCER OK
331 000040 S.HDHM= BIT5 ; HEADS HOME
332 000100 S.BRHM= BIT6 ; BRUSHES HOME
333 000200 S.DOOR= BIT7 ; DOOR INTERLOCKED
334 000400 S.CART= BIT8 ; CARTRAGE INTERLOCK
335 001000 S.SPOK= BIT9 ; SPEED OK
336 002000 S.FWD= BIT10 ; FORWARD
  
```

```

337      004000      S.REV= BIT11      ;REVERSE
338      010000      S.LOAD= BIT12     ;HEADS LOADING
339      020000      S.RTZ= BIT13      ;RETURN TO ZERO
340      040000      S.UNLD= BIT14     ;HEADS UNLOADING
341
342      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
343
344      000020      S.SECT= BIT4      ;SECTOR ERROR
345      000040      S.WCLK= BIT5      ;WRITE CLOCK AND NO WRITE GATE
346      000100      S.WGAT= BIT6      ;WRITE GATE AND NO TRANSISTIONS
347      000200      S.HDFL= BIT7      ;HEAD FAULT
348      000400      S.MHD= BIT8       ;MULTIPLE HEAD SELECT
349      001000      S.XERR= BIT9      ;INDEX ERROR
350      002000      S.DIB= BIT10     ;DIBIT ERROR
351      004000      S.PLO= BIT11     ;PLO ERROR
352      010000      S.NMOV= BIT12    ;SEEK AND NO MOTION
353      020000      S.LIND= BIT13    ;LIMIT DETECT ON SEEK
354      040000      S.BRKE= BIT14    ;SERVO-BRAKE
355
356      .SBTTL  COMMON MASKS
357
358      000007      M.DRV= 7          ;DRIVE CODE
359      100000      M.PAR= BIT15     ;PARITY
360      000003      M.ID= 3          ;BYTE ID
361      017760      M.CDIF= 17760    ;CYLINDER DIFFERENCE/OFFSET
362      017760      M.CADD= 17760    ;CYLINDER ADDRESS
363      077770      M.SER= 77770     ;DRIVE SERIAL NUMBER
364      000760      M.SECT= 760      ;SECTOR COUNT
365      007000      M.HEAD= 7000     ;HEAD DECODE
366
367      .SBTTL  TRAP CATCHER
368
369      000000      .=0
370
371      ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
372      ;*A "+2, IOT" SEQUENCE TO CATCH AND PROCESS ILLEGAL
373      ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
374      ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
375      ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
376      ;*LESS THAN 1002) JUMPS TO THE SERROR ROUTINE.
377      ;*THE SERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
378      ;* PC=YYYYYY UNEXPECTED TRAP TO XXX
379      ;* AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
380      ;* WHERE XXX=LOCATION OF ILLEGAL TRAP
381      ;* YYYYYY=PC AT TIME OF TRAP
382      ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
383      ;* CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.
384
385      $40CAT: HALT      ;: HALT
386      BR      .-100    ;: BRANCH TO 177700 & TIME OUT (NOT ON
387      ;: 11/05)
388      .WORD   START    ;: VECTOR TO STARTING ADDRESS
389      .WORD   340      ;: WITH PRIORITY LEVEL 7
390      .=174
391      DISPREG: .WORD   0      ;: SOFTWARE DISPLAY REGISTER
392      SWREG:  .WORD   0      ;: SOFTWARE SWITCH REGISTER
393
394      .SBTTL  STARTING ADDRES(ES)
395      JMP      @#START ;;GO TO START OF PROGRAM
  
```

```

393 000204 000137 001726      JMP      RESTRT      ;JUMP TO RESTART ROUTINE
394      000214 000137 001716      JMP      PARM        ;JUMP TO OPERATOR ASSIGNED PARMETERS
395      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
396      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
397      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
398      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
399      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
400      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
401      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
402      000220 000137 001706      JMP      SETCLK      ;JUMP TO SET CLOCK ROUTINE
403      000224 000024 000200      .SBTTL  APT PARAMETER BLOCK
404      000024 000024 000200      .SX=.      ;;SAVE CURRENT LOCATION
405      000024 000024 000200      =24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
406      000044 000044 000224      200      ;;FOR APT START UP
407      000044 000044 000224      =44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
408      000044 000044 000224      $APTHDR  ;;POINT TO APT HEADER BLOCK
409      000044 000044 000224      =.SX     ;;RESET LOCATION COUNTER
410      000044 000044 000224      ;;*****
411      000044 000044 000224      ;;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
412      000044 000044 000224      ;;INTERFACE SPEC.
413      000224 000000 000000      $APTHD:
414      000224 000000 000000      $HIBTS: .WORD 0      ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
415      000226 001276 000024      $MADR: .WORD $MAIL   ;; ADDRESS OF APT MAILBOX (BITS 0-15)
416      000230 000024 000074      $TSTM: .WORD 20.     ;; RUN TIM OF LONGEST TEST
417      000232 000074 000740      $PASTM: .WORD 60.    ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
418      000234 000740 000031      $UNITM: .WORD 480.   ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
419      000236 000031 000031      .WORD  $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)
  
```

420  
421  
422  
423  
424  
425  
426  
427 001100  
428 001100 000000  
429 001102 000  
430 001103 000  
431 001104 000000  
432 001106 000000  
433 001110 000000  
434 001112 000000  
435 001114 000  
436 001115 001  
437 001116 000000  
438 001120 000000  
439 001122 000000  
440 001124 000000  
441 001126 000000  
442 001130 000000  
443 001132 000000  
444 001134 000  
445 001135 000  
446 001136 000000  
447 001140 177570  
448 001142 177570  
449 001144 177560  
450 001146 177562  
451 001150 177564  
452 001152 177566  
453 001154 000  
454 001155 002  
455 001156 012  
456 001157 000  
457 001160 000000  
458  
459 001162 000000  
460 001164 000000  
461 001166 000000  
462 001170 000000  
463 001172 000000  
464 001174 000000  
465 001176 000000  
466 001200 000000  
467 001202 000000  
468 001204 000000  
469 001206 000000  
470 001210 000000  
471 001212 000000  
472 001214 000000  
473 001216 000000  
474 001220 000000  
475 001222 000000

.SBTTL COMMON TAGS

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

SCMTAG: .=1100  
STSTNM: .WORD 0  
SERFLG: .BYTE 0  
SICNT: .WORD 0  
SLPADR: .WORD 0  
SLPERR: .WORD 0  
SERTTL: .WORD 0  
SITEMB: .BYTE 0  
SERMAX: .BYTE 1  
SERRPC: .WORD 0  
SGDADR: .WORD 0  
\$BDADR: .WORD 0  
SGDDAT: .WORD 0  
\$BDDAT: .WORD 0  
SAUTOB: .WORD 0  
SINTAG: .BYTE 0  
SWR: .WORD DSWR  
DISPLAY: .WORD DDISP  
\$TKS: 177560  
\$TKB: 177562  
\$TPS: 177564  
\$TPB: 177566  
\$NULL: .BYTE 0  
\$FILLS: .BYTE 2  
\$FILLC: .BYTE 12  
\$TPFLG: .BYTE 0  
\$REGAD: .WORD 0  
\$REG0: .WORD 0  
\$REG1: .WORD 0  
\$REG2: .WORD 0  
\$REG3: .WORD 0  
\$REG4: .WORD 0  
\$REG5: .WORD 0  
\$REG6: .WORD 0  
\$REG7: .WORD 0  
\$REG10: .WORD 0  
\$REG11: .WORD 0  
\$REG12: .WORD 0  
\$REG13: .WORD 0  
\$REG14: .WORD 0  
\$REG15: .WORD 0  
\$REG16: .WORD 0  
\$REG17: .WORD 0  
\$TMP0: .WORD 0

;; START OF COMMON TAGS  
;; CONTAINS THE TEST NUMBER  
;; CONTAINS ERROR FLAG  
;; CONTAINS SUBTEST ITERATION COUNT  
;; CONTAINS SCOPE LOOP ADDRESS  
;; CONTAINS SCOPE RETURN FOR ERRORS  
;; CONTAINS TOTAL ERRORS DETECTED  
;; CONTAINS ITEM CONTROL BYTE  
;; CONTAINS MAX. ERRORS PER TEST  
;; CONTAINS PC OF LAST ERROR INSTRUCTION  
;; CONTAINS ADDRESS OF 'GOOD' DATA  
;; CONTAINS ADDRESS OF 'BAD' DATA  
;; CONTAINS 'GOOD' DATA  
;; CONTAINS 'BAD' DATA  
;; RESERVED--NOT TO BE USED  
;; AUTOMATIC MODE INDICATOR  
;; INTERRUPT MODE INDICATOR  
;; ADDRESS OF SWITCH REGISTER  
;; ADDRESS OF DISPLAY REGISTER  
;; TTY KBD STATUS  
;; TTY KBD BUFFER  
;; TTY PRINTER STATUS REG. ADDRESS  
;; TTY PRINTER BUFFER REG. ADDRESS  
;; CONTAINS NULL CHARACTER FOR FILLS  
;; CONTAINS # OF FILLER CHARACTERS REQUIRED  
;; INSERT FILL CHARS. AFTER A "LINE FEED"  
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
;; CONTAINS THE ADDRESS FROM WHICH (\$REG0) WAS OBTAINED  
;; CONTAINS ((\$REGAD)+0)  
;; CONTAINS ((\$REGAD)+2)  
;; CONTAINS ((\$REGAD)+4)  
;; CONTAINS ((\$REGAD)+6)  
;; CONTAINS ((\$REGAD)+10)  
;; CONTAINS ((\$REGAD)+12)  
;; CONTAINS ((\$REGAD)+14)  
;; CONTAINS ((\$REGAD)+16)  
;; CONTAINS ((\$REGAD)+20)  
;; CONTAINS ((\$REGAD)+22)  
;; CONTAINS ((\$REGAD)+24)  
;; CONTAINS ((\$REGAD)+26)  
;; CONTAINS ((\$REGAD)+30)  
;; CONTAINS ((\$REGAD)+32)  
;; CONTAINS ((\$REGAD)+34)  
;; CONTAINS ((\$REGAD)+36)  
;; USER DEFINED

F04

476	001224	000000	\$TMP1:	.WORD	0	::	USER DEFINED	
477	001226	000000	\$TMP2:	.WORD	0	::	USER DEFINED	
478	001230	000000	\$TMP3:	.WORD	0	::	USER DEFINED	
479	001232	000000	\$TMP4:	.WORD	0	::	USER DEFINED	
480	001234	000000	\$TMP5:	.WORD	0	::	USER DEFINED	
481	001236	000000	\$TMP6:	.WORD	0	::	USER DEFINED	
482	001240	000000	\$TMP7:	.WORD	0	::	USER DEFINED	
483	001242	000000	\$TMP10:	.WORD	0	::	USER DEFINED	
484	001244	000000	\$TMP11:	.WORD	0	::	USER DEFINED	
485	001246	000000	\$TMP12:	.WORD	0	::	USER DEFINED	
486	001250	000000	\$TMP13:	.WORD	0	::	USER DEFINED	
487	001252	000000	\$TMP14:	.WORD	0	::	USER DEFINED	
488	001254	000000	\$TMP15:	.WORD	0	::	USER DEFINED	
489	001256	000000	\$TMP16:	.WORD	0	::	USER DEFINED	
490	001260	000000	\$TMP17:	.WORD	0	::	USER DEFINED	
491	001262	000000	\$TIMES:	0		::	MAX. NUMBER OF ITERATIONS	
492	001264	000000	\$ESCAPE:	0		::	ESCAPE ON ERROR ADDRESS	
493	001266	177607	\$BELL:	.ASCIZ	<207><377><377>	::	CODE FOR BELL	
494	001272	077	\$QUES:	.ASCII	/?/	::	QUESTION MARK	
495	001273	015	\$CRLF:	.ASCII	<15>	::	CARRIAGE RETURN	
496	001274	000012	\$LF:	.ASCIZ	<12>	::	LINE FEED	
497			;:*****					
498			.SBTTL					APT MAILBOX-ETABLE
499			;:*****					
500			.EVEN					
501			\$MAIL:			::	APT MAILBOX	
502	001276		\$MSGTY:	.WORD	AMSGTY	::	MESSAGE TYPE CODE	
503	001276	000000	\$FATAL:	.WORD	AFATAL	::	FATAL ERROR NUMBER	
504	001300	000000	\$TESTN:	.WORD	ATESTN	::	TEST NUMBER	
505	001302	000000	\$PASS:	.WORD	APASS	::	PASS COUNT	
506	001304	000000	\$DEVCT:	.WORD	ADEVCT	::	DEVICE COUNT	
507	001306	000000	\$UNIT:	.WORD	AUNIT	::	I/O UNIT NUMBER	
508	001310	000000	\$MSGAD:	.WORD	AMSGAD	::	MESSAGE ADDRESS	
509	001312	000000	\$MSGLG:	.WORD	AMSGLG	::	MESSAGE LENGTH	
510	001314	000000	\$ETABLE:			::	APT ENVIRONMENT TABLE	
511	001316		\$ENV:	.BYTE	AENV	::	ENVIRONMENT BYTE	
512	001316	000	\$ENVM:	.BYTE	AENVM	::	ENVIRONMENT MODE BITS	
513	001317	000	\$SWREG:	.WORD	ASWREG	::	APT SWITCH REGISTER	
514	001320	000000	\$USWR:	.WORD	AUSWR	::	USER SWITCHES	
515	001322	000000	\$CPUOP:	.WORD	ACPUOP	::	CPU TYPE, OPTIONS	
516	001324	000000	;:*****					
517			BITS 15-11=CPU TYPE					
518			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05					
519			11/70=06, PDQ=07, Q=10					
520			BIT 10=REAL TIME CLOCK					
521			BIT 9=FLOATING POINT PROCESSOR					
522			BIT 8=MEMORY MANAGEMENT					
523	001326	000	\$MAMS1:	.BYTE	AMAMS1	::	HIGH ADDRESS, M.S. BYTE	
524	001327	000	\$MTYP1:	.BYTE	AMTYP1	::	MEM. TYPE, BLK#1	
525			MEM. TYPE BYTE -- (HIGH BYTE)					
526			900 NSEC CORE=001					
527			300 NSEC BIPOLAR=002					
528			500 NSEC MOS=003					
529	001330	000000	\$MADR1:	.WORD	AMADR1	::	HIGH ADDRESS, BLK#1	
530			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE					
531	001332	000	\$MAMS2:	.BYTE	AMAMS2	::	HIGH ADDRESS, M.S. BYTE	

G04

532	001333	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE, BLK#2
533	001334	000000	\$MADR2:	.WORD	AMADR2	::MEM. LAST ADDRESS, BLK#2
534	001336	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS, M.S. BYTE
535	001337	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE, BLK#3
536	001340	000000	\$MADR3:	.WORD	AMADR3	::MEM. LAST ADDRESS, BLK#3
537	001342	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS, M.S. BYTE
538	001343	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE, BLK#4
539	001344	000000	\$MADR4:	.WORD	AMADR4	::MEM. LAST ADDRESS, BLK#4
540	001346	000210	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1, BUS PRIORITY#1
541	001350	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2, BUS PRIORITY#2
542	001352	177440	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
543	001354	000000	\$DEVN:	.WORD	ADEVN	::DEVICE MAP
544	001356	000000	\$CDW1:	.WORD	ACDW1	::CONTROLLER DESCRIPTION WORD#1
545	001360		\$ETEND:			
546			.MEXIT			

547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602

001360

001360 000000  
001362 053670  
001364 055242  
001366 055334

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

;\* EM AND DH ARE ASCIZ DATA. EM IS ALWAYS A MESSAGE BUT DH  
;\* CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACCROSS  
;\* THE PAGE, DT IS A STRING OF WORDS THAT POINT TO THE DATA TO  
;\* BE TYPED, AND DF IS A STRING OF WORK THAT TELL HOW THE DT WORDS  
;\* ARE TO BE TYPED. IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A  
;\* PARTICULAR FORMAT, IT IS REPLACED WITH A ZERO.  
;\* THE NORMAL USAGE OF THE ERROR TABLE IS TO HAVE A TABLE ENTRY FOR  
;\* EACH ERROR MESSAGE THAT CAN OCCUR. IN THE INTEREST OF ECONOMICS  
;\* OF CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE IN A  
;\* SIGHTLY DIFFERENT MANNERS AS DESCRIBED BELOW.

;\* THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT FORMATS THAT ARE USED  
;\* THROUGHOUT THE PROGRAM. WHEN AN ERROR IS TO BE REPORTED, THE  
;\* TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT IS CHOSEN FROM  
;\* THE DEFINED SET. THE TABLE ENTRY CHOSEN IS THEN ALTERED  
;\* BY CHANGING THE FIRST (AND POSSIBLY THE SECOND) WORD TO CONTAIN  
;\* THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE MESSAGE  
;\* PORTION OF THE REPORT. THE DATA FIELDS FOR THAT ENTRY ARE NEVER  
;\* CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.  
;\* THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW WITH  
;\* THE DEFINITION OF THE ENTRY. ALL DATA FIELDS ARE TYPED IN OCTAL.

;.ERROR ITEM 1  
;(MESSAGE)  
;\* TST NUM ERR PC DRIVE  
;\* \$TESTN \$ERRPC DRVNUM

EM1N: .WORD 0  
DH001  
DT001  
DF001

;.ERROR ITEM 2  
;(MESSAGE)  
;\* (MESSAGE)  
;\* TST NUM ERR PC DRIVE  
;\* \$TESTN \$ERRPC DRVNUM  
;\* RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA



603			.*	T.CS1	T.CS2	T.DS	T.ER	T.ASOF	T.DCYL	T.DA
604			.*	RKBA	RKWC					
605			.*	T.BA	T.WC					
606										
607	001370	000000	EM2N:	.WORD	0					
608	001372	000000	DH2N:	.WORD	0					
609	001374	055250		DT002						
610	001376	055340		DF002						
611				.*	ERROR ITEM 3					
612				.*	(MESSAGE)					
613			.*	TST NUM	ERR PC	DRIVE				
614			.*	\$TESTN	\$ERRPC	DRVNUM				
615			.*	RKCS1	RKCS2	RKDS	RKER	RKASOF	RKMR1	
616			.*	T.CS1	T.CS2	T.DS	T.ER	T.AST	T.MR1	
617										
618	001400	000000	EM3N:	.WORD	0					
619	001402	053716	DH002A							
620	001404	055220		DT003						
621	001406	055360		DF003						

.\* ERROR ITEMS 4,5,6,8,7 ARE USED TO REPORT ERRORS THAT ARE THE RESULT  
 .\* OF A HARDWARE ERROR INDICATOR BEING SET WHEN NOT EXPECTED,  
 .\* NOT SET WHEN IT IS EXPECTED, OR BOTH. THE ERROR REPORT WILL  
 .\* CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2) ALL THE EXPECTED  
 .\* ERRORS THAT DID NOT OCCUR, OR (3) ALL THE EXPEDTED BUT NOT SET ERRORS  
 .\* AND THE UNEXPECTED BUT SET ERRORS.

.\* THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT.  
 .\* INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING  
 .\* STATEMENTS:

.\* "THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"  
 .\* "THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"  
 .\* "THE ABOVE ARE ERRORS SET IN OPERATION:"

.\* PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT  
 .\* SPECIFY TJE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE  
 .\* THAT IDENTIFIES THE OPERATION BEING PERFORMED.

.\* EXAMPLE:  
 .\* NON-EXISTANT DRIVE  
 .\* THE ABOVE ARE ERRORS SET IN OPERATION:  
 .\* DRIVE SELECT  
 .\* (ADDITIONAL LINES OF INFORMATION)

.\* THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.

.\* EXAMPLE:  
 .\* NON-EXISTANT DRIVE  
 .\* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:  
 .\* DRIVE SELECT  
 .\* (ADDITIONAL LINES OF INFORMATION)

.\* THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR, I.E.  
 .\* A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.

622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658

659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714

```

001410 000000
001412 000000
001414 055250
001416 055370

001420 000000
001422 000000
  
```

```

: * EXAMPLE:
: * NON-EXISTANT MEMORY
: * THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
: * UNIBUS PARITY ERROR
: * THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
: * WRITE DATA
: * (ADDITIONAL LINES OF INFORMATION)
: *
: * THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN UNIBUS
: * PARITY ERROR WAS EXPECTED.
: *
: * ERROR ITEM 4
: * (DESCRIPTION OF ERROR)
: * ERROR IN OPERATION
: * (DESCRIPTION OF OPERATION)
: * TST NUM ERR PC DRIVE
: * STESTN SERRPC DRVNUM
: * RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
: * T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
: * RKBA RKWC
: * T.BA T.WA
: * A00 B00 A01 B01 A02 B02 A03 B03
: * $REG10 $REG11 $REG12 $REG13 $REG14 $REG15 $REG16 $REG17
: *
: * THE ERRORS REPORTED BY THIS FORMAT ARE:
: * CONTROLLER DETECTED DRIVE BUS ERROR
: * DRIVE DETECTED DRIVE BUS ERROR
: * SEEK INCOMPLETE
: * NON-EXECUTABLE DRIVE FUNCTION
: * DRIVE TIMING ERROR
: * DRIVE UNSAFE
: * AC LOW
: * SPINDLE SPEED LOSS
: * DRIVE OFF TRACK
: * ILLEGAL DRIVE ADDRESS ERROR
: * CYLINDER OVERFLOW
: * DRIVE TYPE ERROR
: * FORMAT ERROR
: * WRITE LOCK ERROR
: *
EM4N: .WORD 0
DH4N: .WORD 0
DT004
DF004

: * ERROR ITEM 5
: * THIS ENTRY IS THE SAME AS ITEM 4 WITH THE ADDITION
: * OF A MESSAGE THAT FOLLOWS. THIS MESSAGE IS:
: *
: * "ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"
: *
: * THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR
: * A00 THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.
: *
EMS5N: .WORD 0
DHS5N: .WORD 0
  
```

715 001424 055250  
716 001426 055420

DT005  
DF005

717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737

\*\*\*  
\*\*\* ERROR ITEM 6  
\*\*\* (DESCRIPTION OF ERROR)  
\*\*\* ERROR IN OPERATION  
\*\*\* (DESCRIPTION OF OPERATION)  
\*\*\* TST NUM ERR PC DRIVE  
\*\*\* STESTN SERRPC DRVNUM  
\*\*\* RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA  
\*\*\* T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA  
\*\*\* RKBA RKWC  
\*\*\* T.BA T.WC  
\*\*\*  
\*\*\* THE ERRORS REPORTED BY THIS FORMAT ARE:  
\*\*\* DATA CHECK  
\*\*\* WRITE CHECK  
\*\*\* ECC HARD  
\*\*\* DATA LATE  
\*\*\* OPERATION INCOMPLETE  
\*\*\* HEADER VRC ERROR  
\*\*\* BAD SECTOR ERROR  
\*\*\*

738 001430 000000  
739 001432 000000  
740 001434 055250  
741 001436 055454

EM6N: .WORD 0  
DH6N: .WORD 0  
DT006  
DF006

742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763

\*\*\*  
\*\*\* ERROR ITEM 7  
\*\*\* (DESCRIPTION OF ERROR)  
\*\*\* ERROR IN OPERATION  
\*\*\* (DESCRIPTION OF OPERATION)  
\*\*\* TST NUM ERR PC DRIVE  
\*\*\* STESTN SERRPC DRVNUM  
\*\*\* RKCS1 RKCS2 RKDS RKER RKASOF  
\*\*\* T.CS1 T.CS2 T.DS T.ER T.ASOF  
\*\*\*  
\*\*\* THE ERRORS REPORTED BY THIS FORMAT ARE:  
\*\*\* NON-EXISTANT DRIVE  
\*\*\* NON-EXISTANT MEMORY  
\*\*\* CONTROLLER TIME OUT  
\*\*\* UNIT FIELD ERROR  
\*\*\* MULTIPLE DRIVE SELECT  
\*\*\* PROGRAMMING ERROR  
\*\*\* UNIBUS PARITY ERROR  
\*\*\* ILLEGAL FUNCTION CODE  
\*\*\*  
\*\*\* DESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL  
\*\*\*

764 001440 000000  
765 001442 000000  
766 001444 055250  
767 001446 055500

EM7N: .WORD 0  
DH7N: .WORD 0  
DT007  
DF007

768  
769  
770

\*\*\*  
\*\*\* ERROR ITEM 10  
\*\*\* (DESCRIPTION OF ERROR)  
\*\*\*

```

771      : *      ERROR AT COMPLETION OF OPERATION
772      : *      (DESCRIPITON OF OPERATION)
773      : *      TST NUM ERR PC DRIVE
774      : *      $TESTN $ERRPC DRVNUM
775      : *      EXPT 15
776      : *      $REG10 $REG11
777
778      : *      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
779      : *      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS. THE SPECIFIC
780      : *      ERRORS ARE;
781      : *      WORD COUNT INCORRECT
782      : *      BUS ADDRESS INCORRECT
783      : *      CYLINDER ADDRESS INCORRECT
784      : *      TRACK ADDRESS INCORRECT
785      : *      SECTOR ADDRESS INCORRECT
786
787      001450 000000      EM10N: .WORD 0
788      001452 054531      DHD10
789      001454 055320      DTO15
790      001456 055520      DFO10
791
792      : *      ERROR ITEM 11
793      : *      (ERROR INDICATOR OR STATUS BIT)
794      : *      NOT SET AS A RESULT OF
795      : *      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
796      : *      TST NUM ERR PC DRIVE
797      : *      $TESTN $ERRPC DRVNUM
798      : *      RKCS1  RKCS2  RKDS  RKER  RKASOF  RKMRI
799      : *      T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.MRI
800
801      001460 000000      EM11N: .WORD 0
802      001462 054655      DHD11
803      001464 055250      DTO10
804      001466 055540      DFO11
805
806      : *      ERROR ITEM 12
807      : *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
808      : *      "NOT RESET AS A RESULT OF"
809
810      001470 000000      EM12N: .WORD 0
811      001472 054704      DHD12
812      001474 055250      DTO10
813      001476 055540      DFO11
814
815      : *      ERROR ITEM 13
816      : *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
817      : *      "SET AS A RESULT OF"
818
819      001500 000000      EM13N: .WORD 0
820      001502 054735      DHD13
821      001504 055250      DTO10
822      001506 055540      DFO11
823
824      : *      ERROR ITEM 14
825      : *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
826      : *      "RESET AS A RESULT OF"
  
```

827 001510 000000  
 828 001512 054760  
 829 001514 055250  
 830 001516 055540  
 831  
 832  
 833  
 834  
 835  
 836  
 837  
 838  
 839 001520 000000  
 840 001522 053716  
 841 001524 055320  
 842 001526 055562  
 843  
 844  
 845  
 846  
 847 001530 000000  
 848 001532 000000  
 849 001534 055326  
 850 001536 055572  
 851

EM14N: .WORD 0  
 DH014  
 DT010  
 DF011

;\* ERROR ITEM 15  
 ;\* (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)  
 ;\* TST NUM ERR PC DRIVE  
 ;\* \$TESTN \$ERRPC DRVNUM  
 ;\* GOOD BAD WORD NUM  
 ;\* \$REG10 \$REG11 \$REG12

EM15N: .WORD 0  
 DH0U2A  
 DT015  
 DF015

;\* ERROR ITEM 16  
 ;\* ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15

0  
 0  
 DT015A  
 DF016

852			.SBTTL	REGISTER STORAGE FOR TEST	
853					
854	001540	000000	T.CS1:	.WORD	0
855	001542	000000	T.WC:	.WORD	0
856	001544	000000	T.BA:	.WORD	0
857	001546	000000	T.DA:	.WORD	0
858	001550	000000	T.CS2:	.WORD	0
859	001552	000000	T.DS:	.WORD	0
860	001554	000000	T.ER:	.WORD	0
861	001556	000000	T.ASOF:	.WORD	0
862	001560	000000	T.DCYL:	.WORD	0
863	001562	000000	T.SPAR:	.WORD	0
864	001564	000000	T.DB:	.WORD	0
865	001566	000000	T.MR1:	.WORD	0
866	001570	000000	T.ECPS:	.WORD	0
867	001572	000000	T.ECPT:	.WORD	0
868	001574	000000	T.MR2:	.WORD	0
869	001576	000000	T.MR3:	.WORD	0
870					
871			.SBTTL	REGISTER SETUP STORAGE	
872	001600	000100	L.CS1:	.WORD	100 ;PRESET WITH INTERRUPT ENABLE
873	001602	000000	L.WC:	.WORD	0
874	001604	000000	L.BA:	.WORD	0
875	001606		L.DA:		
876	001606	000	L.DS:	.BYTE	0
877	001607	000	L.DT:	.BYTE	0
878	001610	000000	L.CS2:	.WORD	0
879	001612	000000	L.ASOF:	.WORD	0
880	001614	000000	L.DCYL:	.WORD	0
881	001616	000000	L.MR1:	.WORD	0
882			.SBTTL	PROGRAM DEFINED VARIABLES	
883					
884	001620	000000	RKVEC:	.WORD	0 ;RK VECTOR
885	001622	000000	RKPRI:	.WORD	0 ;RK PRIORITY
886	001624	000000	SRTFLG:	.WORD	0 ;START FLAG
887					0 = 200
888					1 = 214
889					-1 = 204
890	001626	000000	DRVNUM:	.WORD	0 ;DRIVE UNDER TEST
891	001630	000000	DRVBIT:	.WORD	0 ;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
892	001632	000024	ERRCNT:	.WORD	↑D20 ;ERROR COUNTER TO LIMIT ERROR
893					ERRORS REPORTED IN PROGRAM
894	001634	000024	ERRLMT:	.WORD	↑D20 ;DATA COMPARE ERROR LIMIT
895	001636	055722	BSF24P:	.WORD	BS24 ;POINTER TO BAD SECTORS 24 SECTOR MODE
896					(FACTORY)
897	001640	055576	BSF26P:	.WORD	BS26 ;POINTER TO BAD SECTORS 26 SECTOR MODE
898					(FACTORY)
899	001642	000000	BSS24P:	.WORD	0 ;POINTER TO BAD SECTORS 24 SECT MODE
900					(SOFTWARE)
901	001644	000000	BSS26P:	.WORD	0 ;POINTER TO BAD SECTORS 26 SECTOR MODE
902					(SOFTWARE)
903	001646	000000	BS26CT:	.WORD	0 ;COUNT OF BAD SECTORS 26 SECTOR MODE
904	001650	000000	BS24CT:	.WORD	0 ;COUNT OF BAD SECTORS 24 SECTOR MODE
905	001652	000764	MILCNT:	.WORD	↑D500 ;COUNT TO APPROXIMATE 1 MILL SEC
906	001654	000017	TIMCNT:	.WORD	↑D15 ;COUNTER FOR MYTIME ROUTINE
907	001656	000000	OPTFLG:	.WORD	0 ;OPTION FLAGS

```

908
909          000001          DOTST= BIT0          ;DRIVE 0 TO BE TESTED FLAG
910          000002          MEMSZB= BIT1          ;MEMORY SIZE REPORT FLAG
911          000004          MEMPYB= BIT2          ;MEMORY PARITY REPORT FLAG
912          000010          SRTINS= BIT3          ;START UP INSTRUCTIONS REPORTED FLAG
913          000200          PARBK0= BIT7          ;PARITY OPTION BANK 0
914          000100          PARBK1= BIT6          ;PARITY OPTION BANK 0 INTERLEAVED MEM
915          000200          BSERPT= BIT7          ;BSE HAS BEEN REPORTED
916          000400          FPFMT= BIT8          ;FIRST PASS FORMAT SWITCH
917          100000          LCLKPR= BIT15         ;LINE CLOCK PRESENT
918
919 001660 000000          LCLKTK: .WORD 0          ;LINE CLOCK TICK COUNTER
920 001662 000000          INTSET: .WORD 0          ;NON-ZERO IF RK06 INTERRUPT SINCE LAST
921                                     ;CLEAR
922
923                                     ;
924                                     ; THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
925 001664 000000          DESHLD: .WORD 0          ;DESTINATION HOLD
926 001666 000000          SRCHLD: .WORD 0          ;SOURCE HOLD
927 001670 000000          WRDNUM: .WORD 0          ;WORD NUMBER IN ERROR HOLD
928 001672 000000          WRDCNT: .WORD 0          ;WORDS LEFT IN COMPARE HOLD
929 001674 177546          KWLADD: .WORD 177546       ;KW11-L ADDRESS
930 001676 000100          KWLVEC: .WORD 100          ;KW11-L VECTOR
931 001700 172100          MMCSR1: .WORD 172100       ;MM11 ADDRESS
932 001702 172102          MMCSR2: .WORD 172102       ;MM11 ADDRESS
933 001704 000114          MMVECA: .WORD 114          ;MM11 VECTOR
  
```

```

934
935
936
937
938          .SBTTL PROGRAM SETUP
939
940 001706 012737 000002 001624 SETCLK: MOV    #2,SRTFLG      ;SET START FLAG FOR CLOCK ADJUST
941 001714 000412                BR      START1
942
943 001716 012737 000001 001624 PARM:   MOV    #1,SRTFLG      ;SET START FLAG FOR PARMETER START
944 001724 000406                BR      START1
945
946 001726 012737 177777 001624 RESTRT: MOV   #-1,SRTFLG     ;LOAD START FLAG FOR PARMETER START
947 001734 000402                BR      START1
948
949 001736 005037 001624          START:  CLR    SRTFLG        ;CLEAR START FLAG
950 001742 000005                START1: RESET      ;RESET THE WHOLE SYSTEM
951 001744 012706 001100          MOV    #STACK,SP   ;INITIALIZE STACK POINTER
952 001750 012746 000340          MOV    #PR7,-(SP)  ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
953 001754 012746 001762          MOV    #1$,-(SP)  ;LOAD START OF PROGRAM
954 001760 000002                RTI      ;LOAD PSW
955
956 001762 004737 042122          1$:   JSR    PC,STKINT  ;INITIALIZE KEYBOARD
957          .SBTTL INITIALIZE THE COMMON TAGS
958          ;;CLEAR THE COMMON TAGS (SCMTAG) AREA
959 001766 012706 001100          MOV    #SCMTAG,R6  ;;FIRST LOCATION TO BE CLEARED
960 001772 005026                CLR    (R6)+        ;;CLEAR MEMORY LOCATION
961 001774 022706 001140          CMP    #SWR,R6    ;;DONE?
962 002000 001374                BNE   #-6          ;;LOOP BACK IF NO
963 002002 012706 001100          MOV    #STACK,SP  ;;SETUP THE STACK POINTER
964          ;;INITIALIZE A FEW VECTORS
965 002006 012737 030514 000020  MOV    #SCOPE,#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
966 002014 012737 000340 000022  MOV    #340,#IOTVEC+2 ;;LEVEL 7
967 002022 012737 031530 000030  MOV    #ERROR,#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
968 002030 012737 000340 000032  MOV    #340,#EMTVEC+2 ;;LEVEL 7
969 002036 012737 043762 000034  MOV    #TRAP,#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
970 002044 012737 000340 000036  MOV    #340,#TRAPVEC+2;LEVEL 7
971 002052 012737 043604 000024  MOV    #SPWRDN,#PWRVEC ;;POWER FAILURE VECTOR
972 002060 012737 000340 000026  MOV    #340,#PWRVEC+2 ;;LEVEL 7
973 002066 013737 030120 030112  MOV    SENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
974 002074 005037 001262          CLR    STIMES      ;INITIALIZE NUMBER OF ITERATIONS
975 002100 005037 001264          CLR    ESCAPE      ;CLEAR THE ESCAPE ON ERROR ADDRESS
976 002104 112737 000001 001115  MOVB   #1,SERMAX   ;ALLOW ONE ERROR PER TEST
977 002112 012737 002112 001106  MOV    #.,SLPADR   ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
978 002120 012737 002120 001110  MOV    #.,SLPERR   ;SETUP THE ERROR LOOP ADDRESS
979          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
980          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
981 002126 013746 000004          MOV    #ERRVEC,-(SP) ;;SAVE ERROR VECTOR
982 002132 012737 002166 000004  MOV    #64$,#ERRVEC ;;SET UP ERROR VECTOR
983 002140 012737 177570 001140  MOV    #DSWR,SWR   ;SETUP FOR A HARDWARE SWICH REGISTER
984 002146 012737 177570 001142  MOV    #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
985 002154 022777 177777 176756  CMP    #-1,#SWR   ;TRY TO REFERENCE HARDWARE SWR
986 002162 001012                BNE   66$         ;BRANCH IF NO TIMEOUT TRAP OCCURRED
987          ;AND THE HARDWARE SWR IS NOT = -1
988 002164 000403                BR    65$         ;BRANCH IF NO TIMEOUT
989 002166 012716 002174          64$:  MOV    #65$, (SP) ;;SET UP FOR TRAP RETURN

```



```

990 002172 000002          RTI
991 002174 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
992 002202 012737 000174 001142      MOV #DISPREG,DISPLAY
993 002210 012637 000004 66$: MOV (SP)+,D#ERRVEC ;;RESTORE ERROR VECTOR
994
995 002214 005037 001304          CLR $PASS ;;CLEAR PASS COUNT
996 002220 132737 000200 001317      BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
997 002226 001403          BEQ 67$ ;;YES,USE NON-APT SWITCH
998 002230 012737 001320 001140      MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
999 002236
1000
1001 .SBTTL TYPE PROGRAM NAME
    ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1002 002236 005227 177777          INC #-1 ;;FIRST TIME?
1003 002242 001065          BNE 68$ ;;BRANCH IF NO
1004 002244 022737 030152 000042      CMP #SENDAD,D#42 ;;ACT-11?
1005 002252 001461          BEQ 68$ ;;BRANCH IF YES
1006 002254 104401 002322          TYPE ,69$ ;;TYPE ASCIZ STRING
1007 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
1008 002260 005737 000042          TST D#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
1009 002264 001012          BNE 70$ ;;BRANCH IF YES
1010 002266 123727 001316 000001      CMPB $ENV,#1 ;;ARE WE RUNNING UNDER APT?
1011 002274 001406          BEQ 70$ ;;BRANCH IF YES
1012 002276 023727 001140 000176      CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
1013 002304 001005          BNE 71$ ;;BRANCH IF NO
1014 002306 104406          GTSWR ;;GET SOFT-SWR SETTINGS
1015 002310 000403          BR 71$
1016 002312 112737 000001 001134 70$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
1017 002320 71$:
1018 002320 000436          BR 68$ ;;GET OVER THE ASCIZ
1019 .:69$: .ASCIZ <CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MAINDEC-11-DZR6KB*<CRLF>
1020 002416 68$:
1021 002416 132737 000200 001317      BITB #BIT7,$ENVM ;;TEST IF DO NOT SIZE
1022 002424 001043          BNE 3$ ;;YES - SKIP
1023 002426 004737 030206          JSR PC,$SIZE
1024 002432 023727 030512 000740      CMP $LSTBK,#740 ;;MAKE SURE MEMORY IS SUFFICIENT
1025 002440 103007          BHIS 2$ ;;YES - SKIP
1026 002442 104401 045646          TYPE ,OPRO05 ;;MESSAGE (NOT ENOUGH MEMORY)
1027 002446 012737 000001 030112      MOV #1,$EOPCT ;;FORCE END OF PROGRAM
1028 002454 000137 030064          JMP $EOP
1029 002460 013700 030512 2$: MOV $LSTBK,R0 ;;GET LAST BANK
1030 002464 012701 000006          MOV #6,R1 ;;SET SHIFT COUNT
1031 002470 013703 030510          MOV $LSTAD,R3 ;;GET LAST ADDRESS
1032 002474 005004          CLR R4 ;;CLEAR R4 FOR OVERFLOW
1033 002476 005737 030244          TST $KT11 ;;MEM MANAGE PRESENT?
1034 002502 100005          BPL 23$ ;;NO - SKIP
1035 002504 006300 22$: ASL R0 ;;SHIFT BANK LEFT
1036 002506 005504          ADC R4 ;;ADD IN CARRY
1037 002510 005301          DEC R1 ;;DECREMENT COUNT
1038 002512 001374          BNE 22$ ;;LOOP IF NOT ZERO
1039 002514 050003          BIS R0,R3 ;;SET BANK BITS IN LAST ADDRESS
1040 002516 112737 000001 001327 23$: MOVB #1,$MTYP1 ;;FORCE MEMORY TYPE TO 1
1041 002524 110437 001326          MOVB R4,$MAMS1 ;;STORE UPPER MEMORY ADDRESS
1042 002530 010337 001330          MOV R3,$MADR1 ;;STORE LOWER ADDRESS
1043 002534 032737 000010 001656 3$: BIT #$RTINS,$OPTFLG ;;TEST IF ALREDY REPORTED
1044 002542 001005          BNE 24$ ;;YES - SKIP
1045 002544 104401 046637          TYPE ,OPRO16 ;;TYPE STARTUP INSTRUCTIONS
  
```

# E05

1046	002550	052737	000010	001656		BIS	#SRTINS,OPTFLG	;SET REPORTED FLAG
1047	002556				24\$:			
1048	002556	022737	000001	001624		CMP	#1,SRTFLG	;CHECK IF PARAMETER START
1049	002564	001122				BNE	15\$	;NO, START TESTING
1050	002566	104401	045537		5\$:	TYPE	,OPR001	;TYPE "RK611 BUS ADDRESS ( ) ="
1051	002572	013746	001352			MOV	\$BASE,-(SP)	;SAVE \$BASE FOR TYPEOUT
1052	002576	104402				TYPOC		;GO TYPE--OCTAL ASCII(ALL DIGITS)
1053	002600	104401	045566			TYPE	,OPR002	
1054	002604	104412				RDOCT		;GET VALUE
1055	002606	012637	001222			MOV	(SP)+,\$TMPD	
1056	002612	001407				BEQ	7\$	;CHECK IF <CR>
1057	002614	022737	160000	001222		CMP	#160000,\$TMPD	;CHECK IF IN I/O PAGE
1058	002622	101361				BHI	5\$	
1059	002624	013737	001222	001352		MOV	\$TMPD,\$BASE	;LOAD NEW BUS ADDRESS
1060	002632	104401	045574		7\$:	TYPE	,OPR003	;TYPE "RK611 VECTOR ADDRESS ( ) ="
1061	002636	013746	001346			MOV	\$VECT1,-(SP)	;GET \$VECT1 FOR TYPEOUT
1062	002642	042716	160000			BIC	#160000,(SP)	;CLEAR PRIORITY BITS
1063	002646	104402				TYPOC		
1064	002650	104401	045566			TYPE	,OPR002	
1065	002654	104412				RDOCT		;GET VALUE
1066	002656	012637	001222			MOV	(SP)+,\$TMPD	
1067	002662	001412				BEQ	10\$	;CHECK IF <CR>
1068	002664	022737	001000	001222		CMP	#1000,\$TMPD	
1069	002672	101757				BLOS	7\$	;CHECK IF LEGAL
1070	002674	042737	017777	001346		BIC	#17777,\$VECT1	;CLEAR OLD VECTOR
1071	002702	053737	001222	001346		BIS	\$TMPD,\$VECT1	;LOAD NEW VECTOR ADDRESS
1072	002710	104401	045624		10\$:	TYPE	,OPR004	;TYPE "RK611 PRIORITY ( ) ="
1073	002714	005046				CLR	-(SP)	
1074	002716	113716	001347			MOVB	\$VECT1+1,(SP)	
1075	002722	006216				ASR	(SP)	;SHIFT 5 BITS RIGHT
1076	002724	006216				ASR	(SP)	
1077	002726	006216				ASR	(SP)	
1078	002730	006216				ASR	(SP)	
1079	002732	006216				ASR	(SP)	
1080	002734	104402				TYPOC		
1081	002736	104401	045566			TYPE	,OPR002	
1082	002742	104412				RDOCT		;GET VALUE
1083	002744	012637	001222			MOV	(SP)+,\$TMPD	
1084	002750	001430				BEQ	15\$	;CHECK IF <CR>
1085	002752	022737	000007	001222		CMP	#7,\$TMPD	;CHECK IF LEGAL
1086	002760	103753				BLO	10\$	
1087	002762	022737	000004	001222		CMP	#4,\$TMPD	
1088	002770	101347				BHI	10\$	
1089	002772	006337	001222			ASL	\$TMPD	;SHIFT 5 BITS LEFT
1090	002776	006337	001222			ASL	\$TMPD	
1091	003002	006337	001222			ASL	\$TMPD	
1092	003006	006337	001222			ASL	\$TMPD	
1093	003012	006337	001222			ASL	\$TMPD	
1094	003016	042737	160000	001347		BIC	#160000,\$VECT1+1	;CLEAR OLD PRIORITY
1095	003024	053737	001222	001347		BIS	\$TMPD,\$VECT1+1	;LOAD RK611 PRIORITY
1096	003032	004737	032434		15\$:	JSR	PC,OPTST	;SETUP PARITY CHECK & CLOCK
1097	003036	013700	001346			MOV	\$VECT1,RO	;STORE VECTOR FOR USE
1098	003042	042700	160000			BIC	#160000,RO	;CLEAR PRIORITY BITS
1099	003046	010037	001620			MOV	RO,RKVEC	
1100	003052	012710	032356			MOV	#INTHLR,(RO)	;SETUP INTERRUPT ADDRESS
1101	003056	113737	001347	001622		MOVB	\$VECT1+1,RKPRI	;STORE PRIORITY FOR USE

1102	003064	013746	001622	MOV	RKPRI,-(SP)	;SET PRIORITY
1103	003070	012746	003076	MOV	#16\$,-(SP)	
1104	003074	000002		RTI		
1105	003076					
1106						

16\$:

```

1107 .SBTTL **BASIC INTERFACE AND OPTION TESTS
1108 ;*****
1109 ;*TEST 1 RK611 BASE ADDRESS TEST
1110 ;* CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT
1111 ;* CAUSE A NON-EXISTANT MEMORY TRAP.
1112 ;*****
1113 003076 000004 TST1: SCOPE
1114 003100 012737 000100 001262 MOV #100,STIMES ;DO 100 ITERATIONS
1115 003106 012706 001100 MOV #STACK,SP ;CLEAN OFF STACK
1116 003112 012701 000004 MOV #4,R1 ;SET POINTER TO VECTOR
1117 003116 012146 MOV (R1)+,-(SP) ;STORE OLD VECTOR CONTENTS
1118 003120 011146 MOV (R1)-,(SP)
1119 003122 012701 000004 MOV #4,R1 ;RESET POINTER
1120 003126 012721 032350 MOV #NEXINT,(R1)+ ;SET VECTOR TO NEM TEST HANDLER
1121 003132 012711 000340 MOV #PR7,(R1) ;SET PRIORITY
1122 003136 013702 001352 MOV $BASE,R2 ;SET POINTER TO RK611 BASE ADDRESS
1123 003142 005037 001662 CLR INTSET ;CLEAR INTERRUPT COUNTER
1124 003146 012762 000000 000000 MOV #0,RKCS1(R2) ;WRITE CS1 TO SEE IN NEM WILL SET
1125 003154 000240 NOP
1126 003156 000240 NOP
1127 003160 000240 NOP
1128 003162 005737 001662 TST INTSET ;TEST IF COUNTER IS 0
1129 003166 001411 BEQ 1$ ;YES - SKIP ERROR REPORT
1130 003170 012737 050057 001360 MOV #EM1,EMIN ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
1131 003176 104001 ERROR 1
1132 003200 012737 000001 030112 MOV #1,SEOPCT ;FORCE END OF PROGRAM
1133 003206 000137 030064 JMP SEOP
1134 003212 012701 000006 1$: MOV #6,R1 ;RESTORE VECTOR
1135 003216 012611 MOV (SP)+,(R1)
1136 003220 012641 MOV (SP)+,-(R1)
1137
1138 ;*****
1139 ;*TEST 2 INTERRUPT VECTOR ADDRESS TEST
1140 ;* CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
1141 ;* EXPECTED ADDRESS.
1142 ;*****
1143 003222 000004 TST2: SCOPE
1144 003224 012737 000100 001262 MOV #100,STIMES ;DO 100 ITERATIONS
1145 003232 012762 005000 000010 MOV #CLR,RKCS2(R2) ;CLEAR SUBSYSTEM, SPECIFICALLY TO
1146 ;CLEAR ANY OLD INTERRUPTS
1147 003240 005037 001662 CLR INTSET ;CLEAR INTERRUPT COUNTER
1148 003244 012762 000300 000000 MOV #RDY!IE,RKCS1(R2) ;WRITE CS1 TO FORCE INTERRUPT
1149 003252 000240 NOP
1150 003254 000240 NOP
1151 003256 000240 NOP
1152 003260 005737 001662 TST INTSET ;TEST IF INTERRUPT OCCURRED
1153 003264 001014 BNE 3$ ;YES - SKIP ERROR REPORT
1154 003266 105737 001103 TSTB SERFLG ;TEST IF ERFLG ALREADY SET. IF SO THE
1155 ;INTERRUPT WENT TO THE WRONG VECTOR
1156 ;AND MESSAGE HAS BEEN REPORTED.
1157 003272 001004 BNE 2$ ;THEREFORE - EXIT
1158 003274 012737 050057 001360 MOV #EM1,EMIN ;MESSAGE (NO INTERRUPT)
1159 003302 104001 ERROR 1
1160 003304 012737 000001 030112 2$: MOV #1,SEOPCT ;FORCE PROGRAM TO ABORT
1161 003312 000137 030064 JMP SEOP
1162 003316 3$:

```

1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218

.SBTTL \*\*STATUS VALID TESTS

\*\*\*\*\*

\*TEST 3 SELECT ALL DRIVES

\*\*\*\*\*  
\*  
\* IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT,  
\* DETERMINE WHAT DRIVES ARE ON-LINE BY  
\* SELECTING ALL DRIVES. IF NON-EXISTENT DRIVE REPORTED  
\* MAKE SURE STATUS VALID IS RESET. IF DRIVE  
\* PRESENT MAKE SURE NO ERROR EXISTS, DRIVE  
\* IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.  
\*  
\* IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES  
\* IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE  
\* CYCLED UP, AND STATUS VALID SET.  
\*  
\* IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON  
\* THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM  
\* START (214) WAS USED. IF THE AUTOMATIC START (200)  
\* IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204)  
\* WILL RETAIN THE TEST STATUS OF DRIVE 0.  
\*  
\* IF THE PARAM START IS USED, THE OPERATOR MUST  
\* EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED  
\* OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO  
\* BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME  
\* VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.  
\*  
\* THE DRIVE MUST BE ON-LINE, CYCLED UP, AND WRITE ENABLED.  
\* IF ANY ONE OF THESE CONDITIONS IS NOT TRUE THAT DRIVE  
\* IS NOT TESTED AND IT IS EXPECTED TO BE OFF-LINE. ADDRESSING  
\* THAT DRIVE SHOULD CAUSE NON-EXISTANT DRIVE ERROR.  
\* AT COMPLETION OF THE TEST  
\* A MESSAGE WILL BE GIVEN TO IDENTIFY THE DRIVES TO BE  
\* USED IN TESTING.

NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE  
ANY OTHER TEST THAT FOLLOWS.

\*\*\*\*\*

TST3: SCOPE  
MOV #50.,STIMES ;DO 50. ITERATIONS  
TSSINIT ;CALL SUBSYSTEM CLEAR AND TEST  
ERROR 3  
  
MOV #PRO,-(SP) ;SET PROCESSOR PRIORITY TO ALLOW  
MOV #IS,-(SP) ;RK611 INTERRUPTS  
RTI  
  
IS: MOV RKVEC,R1 ;GET VECTOR  
MOV #INTHLR,(R1)+ ;LOAD INTERRUPT VECTOR  
MOV #PR7,(R1)  
MOV #SDEVN,R3 ;GET ADDRESS OF DEVICE MAP  
TST \$PASS ;TEST IF FIRST PASS  
BNE 7\$ ;NO - SKIP TO DRIVE SELECT TEST

003316 000004  
003320 012737 000062 001262  
003326 104416  
003330 104003  
  
003332 012746 000000  
003336 012746 003344  
003342 000002  
  
003344 013701 001620  
003350 012721 032356  
003354 012711 000340  
003360 012703 001354  
003364 005737 001304  
003370 001104

1219	003372	132737	000200	001317		BITB	#BIT7,SENV	:TEST IF SHOULD SIZE
1220	003400	001402				BEQ	92\$	:YES - SKIP TO DRIVE SIZING.
1221	003402	000137	003734			JMP	11\$	
1222	003406	005013			92\$:	CLR	(R3)	:CLEAR DEVICE MAP
1223	003410	123727	000041	000013		CMPB	#41,#13	:TEST IF RK06 IS LOAD DEVICE
1224	003416	001066				BNE	77\$	:NO - SKIP
1225	003420	022737	000001	001624		CMP	#1,SRTFLG	:WAS START AT PARAM?
1226	003426	001406				BEQ	2\$	:YES - SKIP
1227	003430	104401	046067			TYPE	,OPR007	:NO TEST OF DRIVE 0
1228	003434	042737	000001	001656		BIC	#DOTST,OPTFLG	:DR FLAG - NO TEST DRIVE 0
1229	003442	000457				BR	7\$	
1230	003444	104401	045717		2\$:	TYPE	,OPR006	:MESSAGE - SWAP PACK ON DRIVE OFF LINE.
1231	003450	005037	001610			CLR	L.CS2	:SET TO DRIVE 0
1232	003454	005037	001232			CLR	\$TMP4	:CLEAR FOR USE AS A SWITCH
1233	003460	012737	000101	001600	3\$:	MOV	#SELDRV,L.CS1	:LOAD FOR SELECT
1234	003466	104417				TLOADRK		:LOAD RK & DO SELECT
1235								
1236	003470	104423				TWAT16		:WAIT 16MS FOR COMPLETION
1237	003472	104002				ERROR	2	:NOT DONE ON TIME
1238								
1239	003474	104420				TGETRK		:GET RK REGISTER
1240	003476	032737	100000	001540		BIT	#CERR,T.CS1	:TEST IF CERR
1241	003504	001414				BEQ	5\$	:NO - SKIP
1242	003506	032737	010000	001550		BIT	#NED,T.CS2	:TEST IF NED
1243	003514	001002				BNE	4\$	:YES - SKIP
1244								
1245	003516	104421				TCHKOP		:CHECK THE OPERATION AND REPORT THE ERROR
1246	003520	104004				ERROR	4 ;OR5,6,7	:AFTER THE ERROR IS REPORTED THE TEST
1247								:IS ABORTED
1248	003522	104401	046067		4\$:	TYPE	,OPR007	:TYPE NO TEST OF DRIVE 0
1249	003526	042737	000001	001656		BIC	#DOTST,OPTFLG	:DR FLAG - NO TEST OF DRIVE 0
1250	003534	000422				BR	7\$	:SKIP OVER WAIT FOR PACK MOUNT
1251	003536	005737	001232		5\$:	TST	\$TMP4	:TEST FLAG DRIVE READY HAS RESET
1252	003542	001010				BNE	6\$	:YES - SKIP TO CHECK IF IT IS SET AGAIN
1253	003544	032737	000200	001552		BIT	#DRDY,T.DS	:ELSE CHECK READY
1254	003552	001342				BNE	3\$	:STILL SET - GET STATUS AGAIN
1255	003554	012737	177777	001232		MOV	#-1,\$TMP4	:ELSE SET FLAG TO INDICATE READY WENT LOW
1256	003562	000736				BR	3\$	:GO GET STATUS AGAIN
1257								
1258	003564	032737	000200	001552	6\$:	BIT	#DRDY,T.DS	:TEST IF READY SET AGAIN
1259	003572	001732				BEQ	3\$	:NO - GO GET STATUS AGAIN
1260	003574	052737	000001	001656	7\$:	BIS	#DOTST,OPTFLG	:ELSE SET DRV 0 TEST FLAG.
1261								
1262	003602	005000			7\$:	CLR	RO	:CLEAR FOR DRIVE NUMBER COUNTER
1263	003604	012701	000001			MOV	#1,R1	:SET BIT 0 AS DRIVE SELECTOR
1264								
1265	003610	032737	000001	001656		BIT	#DOTST,OPTFLG	:TEST DRIVE 0?
1266	003616	001430				BEQ	9\$	:NO - SKIP.
1267								
1268	003620	104416			8\$:	TSSINIT		:INITIALIZE SUBSYSTEM
1269	003622	104003				ERROR	3	:ERROR IF NOT SUCCESSFUL
1270								
1271	003624	010037	001610			MOV	RO,L.CS2	:LOAD DRIVE NUMBER
1272	003630	012737	000101	001600		MOV	#SELDRV,L.CS1	:LOAD DRIVE SELECT
1273	003636	104417				TLOADRK		:LOAD RK REGS
1274								

```

1275 003640 104423          TWAT16          ;WAIT FOR INTERRUPT
1276 003642 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
1277
1278 003644 104420          TGETRK          ;GET RK REGS
1279 003646 032737 100000 001540  BIT #CERR,T.CS1 ;ERROR?
1280 003654 001017          BNE 10$         ;YES - SKIP
1281 003656 032737 000200 001552  BIT #DRDY,T.DS ;ELSE TEST IF DRIVE READY
1282 003664 001405          BEQ 9$          ;NO - SKIP
1283 003666 032737 004000 001552  BIT #WRL,T.DS  ;ELSE TEST IF WRITE LOCKED
1284 003674 001001          BNE 9$          ;YES - SKIP
1285
1286 003676 050113          BIS R1,(R3) ;SET BIT - DRIVE PRESET IN MAP
1287
1288 003700 005200          9$: INC RO         ;BUMPS TO NEXT DRIVE
1289 003702 006301          ASL R1          ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
1290 003704 032701 000400  BIT #BIT8,R1   ;WAS LAST DRIVE DONE?
1291 003710 001743          BEQ 8$          ;YES - SKIP
1292 003712 000410          BR 11$         ;ELSE LOOP TO SELECT NEXT DRIVE
1293
1294 003714 032737 010000 001550 10$: BIT #NED,T.CS2 ;WAS CERR DUE TO NED?
1295 003722 001366          BNE 9$          ;YES - BUMP TO NEXT DRIVE
1296
1297 003724 104421          TCHKOP          ;ELSE REPORT THE ERRORS
1298 003726 104004          ERROR 4 ;OR5,6,7
1299 003730 000000          101$: .WORD 0
1300 003732 177777          100$: .WORD -1
1301 003734 005737 003732 11$: TST 100$       ;A SWITCH - IT NEVER GETS EXECUTED
1302 003740 100041          BPL 16$         ;TEST SWITCH
1303 003742 005237 003732          INC 100$        ;IF PLUS - SKIP DRIVE TEST MESSAGE
1304
1305 003746 005713          TST (R3)        ;ANY DRIVE AVAILABLE?
1306 003750 001007          BNE 12$         ;BR IF NOT ZERO
1307 003752 104401 046157  TYPE ,OPRO08   ;ELSE REPORT NO DRIVES AVAILABLE
1308 003756 012737 000001 030112  MOV #1,SEOPCT  ;FORCE PROGRAM ABORT
1309 003764 000137 030064          JMP SEOP        ;GO TO END OF PASS
1310
1311 003770 012701 000200          12$: MOV #BIT7,R1  ;SET DRIVE SELECTOR FOR DRIVE 7
1312 003774 012700 000007          MOV #7,RO       ;SET DRIVE NUMBER TO 7
1313 004000 104401 046242          TYPE ,OPRO09   ;TYPE PREFIX TO DRIVE TEST MESSAGE
1314
1315 004004 030113          13$: BIT R1,(R3) ;TEST IF THIS DRIVE TO BE TESTED
1316 004006 001004          BNE 15$         ;YES - SKIP
1317
1318 004010 005300          14$: DEC RO         ;ELSE DECREMENT DRIVE NUMBER
1319 004012 006201          ASR R1          ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
1320 004014 001373          BNE 13$         ;IF NOT SHIFTED OUT - LOOP
1321 004016 000412          BR 16$         ;ELSE GO TO STATUS VALID TEST
1322
1323 004020 010037 003730          15$: MOV RO,101$   ;PUT DRIVE NUMBER IN TYPE LOCATION
1324 004024 052737 000060 003730  BIS #BIT4:BIT5,101$ ;MAKE IT ASCIZ
1325 004032 104401          TYPE           ;TYPE IT
1326 004034 003730          101$
1327 004036 104401 045534          TYPE SPACE2    ;TYPE SOME SPACES
1328 004042 000762          BR 14$         ;LOOP
1329
1330 004044 005000          16$: CLR RO         ;CLEAR DRIVE NUMBER COUNTER
    
```

## K05

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(732) 01-OCT-76 10:50 PAGE 29  
 DZR6KB.P11 T3 SELECT ALL DRIVES

SEQ 0062

```

1331 004046 012701 000001             MOV     #1,R1          ;SET DRIVE SELECTOR TO DRIVE 0
1332 004052 012737 177777 001240       MOV     #-1,$TMP7      ;SET $TMP7 NEGATIVE
1333 004060 012737 177777 001630       MOV     #-1,DRVBIT    ;SET DRIVE SELECT BIT NEGATIVE
1334
1335 004066 104416             17$:   TSSINIT           ;CLEAR SUBSYSTEM
1336 004070 104003             ERROR   3             ;ERROR FOR BAD CLEAR
1337
1338 004072 010037 001610             MOV     RO,L.CS2      ;SET DRIVE SELECT
1339 004076 010037 001626             MOV     RO,DRVNUM     ;SET DRIVE NUMBER
1340 004102 012737 000101 001600       MOV     #SELDRV,L.CS1 ;SET FOR DRIVE SELECT
1341
1342 004110 104417             TLOADRK           ;LOAD RK REGS
1343 004112 104423             TWAT16           ;WAIT FOR INTERRUPT
1344 004114 104002             ERROR   2           ;ERROR TO SLOW/NOT COMPLETE
1345 004116 030113             BIT     R1,(R3)      ;WAS THAT DRIVE AVAILABLE
1346 004120 001023             BNE    19$         ;YES - SKIP
1347
1348 004122 104422             TCHKWE           ;CHECK THAT ERROR OCCURRED
1349 004124 000000             .WORD   0           ;NONE OF GROUP 1
1350 004126 000000             .WORD   0           ;NONE OF GROUP 2
1351 004130 000001             .WORD   1           ;NED IN GROUP 3
1352 004132 104004             ERROR   4 ;OR5,6,7 ;ERROR IF NO ERROR OR WRONG ERROR
1353
1354 004134 032737 100000 001552       BIT     #SVAL,T.DS    ;DID STATUS VALID RESET?
1355 004142 001404             BEQ    18$         ;YES - SKIP
1356 004144 012737 052201 001400       MOV     #EM47,EM3N    ;ELSE MESSAGE (SVAL NOT RESET WITH NED)
1357 004152 104003             ERROR   3
1358 004154 005200             18$:   INC     RO         ;BUMP TO NEXT DRIVE
1359 004156 006301             ASL    R1         ;SHIFT DRIVE SELECT BIT
1360 004160 032701 000400             BIT     #BIT8,R1     ;ALL DRIVES CHECKED
1361 004164 001740             BEQ    17$         ;NO - GET NEXT DRIVE
1362 004166 000457             BR     21$         ;ELSE EXIT
1363 004170 104421             19$:   TCHKOP           ;CHECK NO ERRORS SET
1364 004172 104004             ERROR   4 ;OR5,6,7 ;REPORT ALL ERRORS
1365
1366 004174 032737 100000 001552       BIT     #SVAL,T.DS    ;CHECK SVAL SET
1367 004202 001004             BNE    20$         ;YES - SKIP
1368 004204 012737 052262 001400       MOV     #EM48,EM3N    ;MESSAGE (NO SVAL FROM EXISTANT DR)
1369 004212 104003             ERROR   3
1370
1371 004214 012737 000103 001600 20$:   MOV     #PACK,L.CS1   ;ELSE SET TO DO PACK ACK
1372 004222 104417             TLOADRK           ;LOAD RK
1373
1374 004224 104423             TWAT16           ;WAIT FOR INTERRUPT
1375 004226 104002             ERROR   2           ;TO SLOW/NOT COMPLETE ERROR
1376
1377 004230 104421             TCHKOP           ;CHECK FOR ANY ERRORS
1378 004232 104004             ERROR   4 ;OR5,6,7 ;YES - REPORT & ABORT TEST
1379
1380 004234 032737 000100 001552       BIT     #VV,T.DS     ;DID VV SET
1381 004242 001005             BNE    22$         ;YES - SKIP
1382 004244 012737 052144 001400       MOV     #EM46,EM3N    ;MESSAGE (VV DID NOT SET/
1383 004252 104003             ERROR   3
1384 004254 000737             BR     18$
1385
1386 004256 032737 040000 001552 22$:   BIT     #DSC,T.DS    ;TEST IF DSC RESET

```



```

1387 004264 001410          BEQ      23$          ;YES - SKIP
1388 004266 012737 053236 001470    MOV     #EMDSC,EM12N
1389 004274 012737 053476 055544    MOV     #EMSCLR,DF011A
1390 004302 104003          ERROR   3           ;"DSC NOT RESET RESULT OF SUBSYS CLEAR"
1391 004304 000723          BR      18$
1392
1393 004306 005737 001630          23$:   TST     DRVBIT          ;TEST IF DRVBIT IS NEGATIVE
1394 004312 100320          BPL     18$          ;NO - SKIP
1395 004314 010137 001630          MOV     R1,DRVBIT      ;STORE DRIVE SELECT BIT
1396 004320 010037 001240          MOV     R0,$TMP7       ;STORE DRIVE NUMBER TO BE TESTED
1397 004324 000713          BR      18$
1398
1399 004326 013737 001240 001626 21$:   MOV     $TMP7,DRVNUM    ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
1400
1401 004334 023727 001624 000002    CMP     SRTFLG,#2      ;TEST IF CLOCK ADJUST START
1402 004342 001002          BNE     25$          ;NO - SKIP
1403 004344 000137 041062          JMP     ADJCLK         ;GO TO ADJUST CLOCK ROUTINE
1404
1405 004350          25$:
1406
1407          ;*****
1408          ;*TEST 4          RELEASE ALL DRIVES
1409          ;*
1410          ;*          RELEASE ALL DRIVES. MAKE SURE NO ERROR
1411          ;*          SETS AND STATUS VALID IS RESET.
1412          ;*
1413          ;*****
1414 004350 000004          TST4:  SCOPE
1415 004352 012737 000062 001262    MOV     #50,$TIMES    ;;DO 50. ITERATIONS
1416 004360 104416          TSSINIT
1417 004362 104003          ERROR   3           ;INITIALIZE SUBSYSTEM
1418          ;BAD INIT
1419 004364 013737 001626 001610    MOV     DRVNUM,L.CS2   ;SET DRIVE NUMBER
1420 004372 012737 000101 001600    MOV     #SELDRV,L.CS1 ;SET DRIVE SELECT
1421
1422 004400 104417          TLOADRK          ;LOAD RK REGS
1423 004402 104423          TWAT16          ;WAIT FOR INTERRUPT
1424 004404 104002          ERROR   2           ;TO SLOW/NOT COMPLETE ERROR
1425
1426 004406 104421          TCHKOP          ;CHECK FOR ANY ERRORS
1427 004410 104004          ERROR   4 ;OR5,6,7 ;REPORT ANY ERRORS
1428
1429 004412 012737 000010 001610    MOV     #RLS,L.CS2    ;SET DRIVE RELEASE,STILL SET FOR SELECT
1430
1431 004420 104417          TLOADRK          ;LOAD RK REGS
1432 004422 104423          TWAT16          ;WAIT FOR INTERRUPT
1433 004424 104002          ERROR   2           ;TO SLOW/NOT COMPLETE ERROR
1434
1435 004426 104421          TCHKOP          ;CHECK FOR ANY ERRORS
1436 004430 104004          ERROR   4 ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
1437 004432 032737 100000 001552    BIT     #SVAL,T.DS    ;DID SVAL RESET?
1438 004440 001404          BEQ     1$          ;YES - SKIP
1439 004442 012737 052341 001400    MOV     #EM49,EM3N    ;MESSAGE (SVAL NOT RESET W/RELEASE)
1440 004450 104003          ERROR   3
1441
1442          1$:

```

1443 004452  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454 004452 000004  
1455 004454 012737 000062 001262  
1456 004462 104416  
1457 004464 104003  
1458 004466 012701 000001  
1459 004472 013737 001626 001610  
1460 004500 012737 000101 001600  
1461 004506 005037 001616  
1462 004512 104417  
1463 004514 104423  
1464 004516 104002  
1465 004520 104421  
1466 004522 104004  
1467  
1468 004524 032737 100000 001552  
1469 004532 001007  
1470  
1471 004534 012737 053076 001460  
1472 004542 012737 045134 055544  
1473 004550 104011  
1474  
1475 004552 010137 001616  
1476  
1477 004556 104417  
1478 004560 104423  
1479 004562 104002  
1480  
1481 004564 104421  
1482 004566 104004  
1483  
1484 004570 032737 100000 001552  
1485 004576 001407  
1486  
1487 004600 012737 053076 001470  
1488 004606 012737 053113 055544  
1489 004614 104012  
1490  
1491 004616 022701 000003  
1492 004622 001402  
1493 004624 005201  
1494 004626 000727  
1495 004630  
1496  
1497  
1498

TSTLUP:  
\*\*\*\*\*  
\*TEST 5 NON-STANDARD MESSAGES AND SVAL  
\*  
\* PICK ONE OF THE AVAILABLE DRIVES AND GET  
\* NON-STANDARD MESSAGES. MAKE SURE NO  
\* ERROR OCCURS AND STATUS VALID DOES NOT SET  
\* AND THAT NON-STANDARD MESSAGES CAUSE STATUS  
\* VALID TO RESET.  
\*  
\*\*\*\*\*

TST5: SCOPE  
MOV #50.,\$TIMES ;DO 50. ITERATIONS  
TSSINIT ;CLEAR SUBSYSTEM  
ERROR 3 ;BAD CLEAR MESSAGE  
MOV #1,R1 ;PRESET R1 FOR MESSAGE PAIR 1  
MOV DRVNUM,L.CS2 ;LOAD DRV NUMBER  
MOV #SELD, L.CS1 ;LOAD SELECT COMMAND  
1\$: CLR L.MR1 ;LOAD FOR STANDARD STATUS  
TLOADRK ;LOAD RK  
TWTAT16 ;WAIT FOR INTERRUPT  
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR  
TCHKOP ;CHECK OPERATION  
ERROR 4 ;5,6 OR 7 ;REPORT ALL ERRORS  
  
BIT #SVAL,T.DS ;TEST STATUS VALID SET  
BNE 2\$ ;YES-SKIP  
  
MOV #EMSVAL,EM11N  
MOV #EMSELD,DF011A  
ERROR 11 ;"SVAL NOT SET RESULT OF DRIVE SELECT"  
  
2\$: MOV R1,L.MR1 ;LOAD MESSAGE PAIR SELECT  
  
TLOADRK ;LOAD RK  
TWTAT16 ;WAIT FOR INTERRUPT  
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR  
  
TCHKOP ;CHECK OPERATION  
ERROR 4 ;5,6, OR 7 ;REPORT ALL ERRORS  
  
BIT #SVAL,T.DS ;TEST STATUS VALID RESET  
BEQ 3\$ ;YES-SKIP  
  
MOV #EMSVAL,EM12N  
MOV #EMNZPR,DF011A  
ERROR 12 ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"  
  
3\$: CMP #3,R1 ;WAS PAIR 3 SELECTED?  
BEQ 4\$ ;YES-SKIP  
INC R1 ;BUMP TO NEXT PAIR  
BR 1\$ ;SKIP TO DO IT.

4\$:  
\*\*\*\*\*  
\*TEST 6 WRITING CS2 AND STATUS VALID  
\*  
\*\*\*\*\*

# N05

```

1499          ;*      SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS
1500          ;*      VALID IS SET.  WRITE COMMAND AND STATUS REGISTER 2.
1501          ;*      MAKE SURE STATUS VALID RESETS.
1502          ;*
1503          ;*****
1504 004630 000004          TST6: SCOPE
1505 004632 012737 000062 001262  MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
1506 004640 104416          TSSINIT          ;CLEAR SUBSYSTEM
1507 004642 104003          ERROR      3          ;BAD INIT ERROR
1508
1509 004644 013737 001626 001610  MOV      DRVNUM,L.CS2     ;LOAD DRIVE NUMBER
1510 004652 012737 000101 001600  MOV      #SELDLV,L.CS1   ;LOAD DRIVE SELECT
1511
1512 004660 104417          TLOADRK          ;LOAD RK
1513 004662 104423          TWAT16          ;WAIT FOR INTERRUPT
1514 004664 104002          ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
1515
1516 004666 104421          TCHKOP          ;CHECK OPERATION
1517 004670 104004          ERROR      4 ;5,6, OR 7 ;REPORT ALL ERRORS
1518
1519 004672 032737 100000 001552  BIT      #SVAL,T.DS     ;TEST STATUS VALID SET
1520 004700 001007          BNE      1$           ;YES-SKIP
1521
1522 004702 012737 053076 001460  MOV      #EMSVAL,EM11N
1523 004710 012737 045134 055544  MOV      #EMSELD,DF011A
1524 004716 104011          ERROR      11          ;"SVAL NOT SET RESULT OF DRV SELECT"
1525
1526 004720 013762 001626 000010 1$:  MOV      DRVNUM,RKCS2(R2) ;WRITE CS2 TO RESET SVAL
1527
1528 004726 104420          TGETRK          ;GET RK REGS.
1529
1530 004730 032737 100000 001552  BIT      #SVAL,T.DS     ;TEST SVAL RESET
1531 004736 001407          BEQ      2$           ;YES-SKIP
1532
1533 004740 012737 053076 001470  MOV      #EMSVAL,EM12N
1534 004746 012737 053154 055544  MOV      #EMWCS2,DF011A
1535 004754 104012          ERROR      12          ;"SVAL NOT RESET BY WRITING CS2"
1536          2$:
1537

```

.SBTTL \*\*CONTROLLER ERROR TESTS

1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593

\*\*\*\*\*  
\*TEST 7 DRIVE TYPE ERROR  
\*  
\* CREATE A DRIVE TYPE ERROR. MAKE SURE DRIVE  
\* TYPE ERROR SETS AND STATUS VALID SETS.  
\*  
\*\*\*\*\*

ST7: SCOPE  
MOV #50.,\$TIMES ;:DO 50. ITERATIONS  
TSSINIT ;:CLEAR SUBSYSTEM  
ERROR 3 ;:BAD INIT ERROR  
  
MOV DRVNUM,L.CS2 ;:LOAD DRIVE NUMBER  
MOV #SELDIV,L.CS1 ;:LOAD DRIVE SELECT  
BIS #CDT,L.CS1 ;:LOAD DRIVE TYPE  
  
TLOADRK ;:LOAD RK  
TWTAT16 ;:WAIT FOR INTERRUPT  
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR  
  
TCHKWE ;:CHECK OPERATION WITH EXPECTED ERROR  
.WORD 000040 ;:DRIVE TYPE ERROR  
.WORD 0  
.WORD 0  
ERROR 4 ; OR 5,6,7 ;:REPORT ANY DIFFERENCES (NO ERRORS,  
;:ADDITIONAL ERRORS, DIFFERENT ERRORS)  
BIT #SVAL, T.DS ;:TEST IF SVAL SET  
BNE IS ;:YES-SKIP  
  
MOV #EMSVAL,EM11N  
MOV #EMDTPE,DF011A  
ERROR 11 ;:"SVAL NOT SET RESULT OF DRV TYPE ERR"

IS:  
\*\*\*\*\*  
\*TEST 10 STATUS VALID AND PARITY ERROR  
\*  
\* ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.  
\* MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION,  
\* DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT,  
\* AND STATUS VALID SET, ISSUE A CONTROLLER  
\* CLEAR. MAKE SURE DRIVE INTERRUPT AND ATTENTION  
\* ARE STILL SET. SELECT DRIVE AGAIN WITH GOOD  
\* PARITY. MAKE SURE ATTENTION, DRIVE STATUS  
\* CHANGE, DRPAR, CONTROLLER ERROR, DRIVE INTERRUPT,  
\* AND STATUS VALID ARE SET AND SPAR IS RESET.  
\* ISSUE A CONTROLLER CLEAR. GET NON-STANDARD MESSAGES  
\* AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION  
\* ARE SET. CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT  
\* FOR ALL AVAILIABLE DRIVES.  
\*  
\*\*\*\*\*

ST10: SCOPE  
MOV #50.,\$TIMES ;:DO 50. ITERATIONS  
TSSINIT ;:CLEAR SUBSYSTEM

1594	005074	104003				ERROR 3		;BAD INIT ERROR
1595								
1596	005076	013737	001626	001610		MOV	DRVNUM,L.CS2	;LOAD DRIVE NUMBER
1597	005104	012737	000101	001600		MOV	#SELDRV,L.CS1	;LOAD DRIVE SELECT
1598	005112	012737	000020	001616		MOV	#PAT,L.MR1	;LOAD EVEN PARITY BIT
1599								
1600	005120	104417				TLOADRK		;LOAD RK REGS-SELECT W/EVEN PARITY
1601	005122	104423				TWAT16		;WAIT FOR INTERRUPT
1602	005124	104002				ERROR 2		;TO SLOW/NOT COMPLETE ERROR
1603								
1604	005126	104422				TCHKWE		;CHECK OPERATION FOR EXPECTED ERROR
1605	005130	000011				DRPARERR:SPARERR		;DRIVE SELECTED DRIVE BUS PARITY ERROR
1606	005132	000000				.WORD 0		;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
1607	005134	000000				.WORD 0		
1608	005136	104004				ERROR 4 ; OR 5,6,7		;REPORT ANY DIFFERENCES
1609								
1610	005140	012700	000400			MOV	#BITB,RO	;ROUTINE TO DETERMINE WHICH BIT
1611								
1612	005144	013701	001626			MOV	DRVNUM,R1	;SHOULD BE SET IN ASOF TO INDICATE
1613	005150	001403				BEQ	3\$	;DRIVE ATTENTION. RO WILL HAVE THE
1614	005152	006300			2\$:	ASL	RO	;BIT THAT SHOULD BE SET FOR THE DRIVE
1615	005154	005301				DEC	R1	;IN USE
1616	005156	001375				BNE	2\$	
1617								
1618	005160	030037	001556		3\$:	BIT	RO,T.ASOF	;TEST ATTENTION SET
1619	005164	001007				BNE	4\$	;YES-SKIP
1620	005166	012737	053262	001460		MOV	#EMDA,EM11N	
1621	005174	012737	051527	055544		MOV	#EMDPA,DF011A	
1622	005202	104011				ERROR	11	; "DRV ATT NOT SET RESULT OF DRV PARITY ERR"
1623	005204	032737	040000	001540	4\$:	BIT	#DI,T.CS1	;TEST DRIVE INTERRUPT SET
1624	005212	001007				BNE	5\$	;YES-SKIP
1625	005214	012737	053216	001460		MOV	#EMDI,EM11N	
1626	005222	012737	051527	055544		MOV	#EMDPA,DF011A	
1627	005230	104011				ERROR	11	; "DRV INT NOT SET RESULT OF DRV PARITY ERR"
1628								
1629	005232	032737	040000	001552	5\$:	BIT	#DSC,T.DS	;TEST DRIVE STATUS CHANGE SET
1630	005240	001007				BNE	6\$	;YES-SKIP
1631	005242	012737	053236	001460		MOV	#EMDSC,EM11N	
1632	005250	012737	051527	055544		MOV	#EMDPA,DF011A	
1633	005256	104011				ERROR	11	; "DSC NOT SET RESULT OF DRV PARITY ERR"
1634								
1635	005260	032737	100000	001552	6\$:	BIT	#SVAL,T.DS	;TEST STATUS VALID SET
1636	005266	001007				BNE	7\$	;YES-SKIP
1637	005270	012737	053076	001460		MOV	#EMSVL,EM11N	
1638	005276	012737	051527	055544		MOV	#EMDPA,DF011A	
1639								
1640	005304	104011				ERROR	11	; "SVAL NOT SET RESULT OF DRV PAR ERR"
1641								
1642	005306	005037	001616		7\$:	CLR	L.MR1	;CLEAR PAT IN MR1
1643								
1644	005312	052737	100000	001600		BIS	#CCLR,L.CS1	;CLEAR CONTROLLER
1645	005320	104417				TLOADRK		;LOAD RK REGS TO DO CLEAR
1646								
1647								
1648	005322	104421				TCHKOP		;CHECK NO ERRORS SET
1649	005324	104004				ERROR 4 ; OR 5,6,7		;REPORT ALL ERRORS STILL SET



```

1706 005576 010137 001616      14$:  MOV      R1,L.MR1      ;LOAD STATUS PAIR SELECTION
1707 005602 104417              TLOADRK              ;LOAD RK REGS
1708 005604 104423              TWAT16              ;WAIT FOR INTERRUPT
1709 005606 104002              ERROR      2        ;TO SLOW/NOT COMPLETE ERROR
1710
1711 005610 104421              TCHKOP              ;CHECK IF ANY ERRORS SET
1712 005612 104004              ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS SET.
1713
1714 005614 030037 001556      BIT      RD,T.ASOF    ;TEST ATTENTION STILL SET
1715 005620 001007              BNE      15$        ;YES-SKIP
1716 005622 012737 053262 001410      MOV      #EMDA,EM4N
1717 005630 012737 053113 055544      MOV      #EMNZPR,DF011A
1718 005636 104014              ERROR      14      ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
1719
1720 005640 032737 040000 001540      15$:  BIT      #DI,T.CS1
1721 005646 001007              BNE      16$
1722 005650 012737 053216 001510      MOV      #EMDI,EM14N
1723 005656 012737 053113 055544      MOV      #EMNZPR,DF011A
1724 005664 104014              ERROR      14      ;"DRV INT RESET RESULT OF NON-0 PAIR SELECT"
1725
1726 005666 005201              16$:  INC      R1          ;BUMP PAIR SELECT
1727 005670 022701 000004      CMP      #4,R1      ;ALL PAIRS DONE?
1728 005674 001340              BNE      14$        ;NO-LOOP
1729
1730 005676 005037 001616      CLR      L.MR1      ;CLEAR MR1
1731
1732 005702 012737 000105 001600      MOV      #CLEAR,L.CS1 ;LOAD DRIVE CLEAR
1733
1734 005710 104417              TLOADRK              ;DO DRIVE CLEAR
1735 005712 104423              TWAT16              ;WAIT FOR INTERRUPT
1736 005714 104002              ERROR      2        ;TO SLOW/NOT COMPLETE ERROR
1737
1738 005716 104421              TCHKOP              ;CHECK FOR ANY ERRORS
1739 005720 104004              ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS
1740
1741 005722 012701 000020              17$:  MOV      #20,R1      ;SET COUNT FOR SHORT WAIT
1742 005726 005301              DEC      R1          ;TO ALLOW CONTROLLER TIME TO POLL
1743 005730 001376              BNE      17$        ;DRIVES
1744
1745 005732 104420              TGETRK              ;GET RK REGS
1746 005734 030037 001556      BIT      RD,T.ASOF    ;TEST ATTENTION RESET
1747 005740 001407              BEQ      18$        ;YES-SKIP
1748 005742 012737 053262 001470      MOV      #EMDA,EM12N
1749 005750 012737 045162 055544      MOV      #EMDCLR,DF011A
1750 005756 104012              ERROR      12      ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
1751
1752 005760 032737 040000 001540      18$:  BIT      #DI,T.CS1
1753 005766 001407              BEQ      19$
1754 005770 012737 053216 001470      MOV      #EMDI,EM12N
1755 005776 012737 045162 055544      MOV      #EMDCLR,DF011A
1756 006004 104012              ERROR      12      ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
1757
1758 006006              19$:
1759 :*****
1760 :*TEST 11      UNIT FIELD ERROR ON RELEASE
1761 :*

```

```

1762
1763
1764
1765
1766
1767
1768
1769 006006 000004
1770 006010 012737 000062 001262
1771 006016 104416
1772 006020 104002
1773
1774 006022 013737 001626 001610
1775 006030 012737 000101 001600
1776
1777 006036 104417
1778 006040 104423
1779 006042 104002
1780
1781 006044 104421
1782 006046 104004
1783
1784 006050 052737 000010 001610
1785 006056 012737 000040 001616
1786
1787 006064 104417
1788
1789 006066 004437 033600
1790 006072 023
1791 006073 002
1792
1793 006074 042762 000040 000026
1794
1795 006102 104423
1796 006104 104002
1797
1798 006106 104422
1799 006110 000000
1800 006112 000000
1801 006114 000004
1802 006116 104004
1803
1804 006120 104416
1805 006122 104002
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817 006124 000004

```

```

*****
TST11: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 2 ;BAD INIT ERROR

MOV DRVNUM,L.CS2 ;SELECT A DRIVE
MOV #SELDRV,L.CS1 ;DO DRIVE SELECT

TLOADRK ;LOAD RK
TWTAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK FOR ANY ERRORS
ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS.

BIS #RLS,L.CS2 ;LOAD RELEASE
MOV #DMD,L.MR1 ;SET DIAGNOSTIC MODE

TLOADRK ;LOAD RK

JSR R4,MCLOCK ;CALL MAINT CLOCK
.BYTE #D19 ;NUMBER OF PHASES
.BYTE 2 ;NUMBER OF CLOCK XISTIONS

BIC #DMD,RKMR1(R2) ;CLEAR DIAG MODE

TWTAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETED

TCHKWE ;CHECK OPERATION WITH ERROR
.WORD 0
.WORD 0
.WORD UFERR ;UNIT FIELD ERROR
ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES

TSSINIT ;CLEAR SUBSYSTEM TO INSURE UFE RESETS
ERROR 2

*****
TST12: SCOPE
UNIT FIELD ERROR ON SELECT

*****
ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE
SELECTED = 0. CLOCK THROUGH PHASE ADDRESS 6.
TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD
ERROR SETS.
*****

```



```

1818 006126 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
1819 006134 104416                      TSSINIT                    ;CLEAR SUBSYSTEM
1820 006136 104003                      ERROR    3                ;BAD INIT ERROR
1821
1822 006140 013737 001626 001610      MOV      DRVNUM,L.CS2     ;LOAD DRIVE NUMBER
1823 006146 012737 000101 001600      MOV      #SELDIV,L.CS1   ;LOAD DRIVE SELECT
1824
1825 006154 104417                      TLOADRK                    ;LOAD RK
1826 006156 104423                      TWAT16                      ;WAIT FOR INTERRUPT
1827 006160 104002                      ERROR    2                ;TO SLOW/NOT COMPLETE
1828
1829 006162 104421                      TCHKOP                      ;CHECK FOR ANY ERROR
1830 006164 104004                      ERROR    4 ; OR 5,6,7    ;REPORT ALL ERRORS
1831
1832 006166 012737 000043 001616      MOV      #DMD!BIT1!BIT0,L.MR1 ;LOAD DIAG MODE & MSG PAIR 3
1833 006174 005037 001610                      CLR      L.CS2            ;LOAD FOR DRIVE 0
1834
1835 006200 104417                      TLOADRK                    ;LOAD RK
1836
1837 006202 004437 033600      JSR      R4,MCLOCK        ;CALL MAINTENANCE CLOCK
1838 006206          026                      .BYTE   1D22              ;THROUGH PHASE 6
1839 006207          002                      .BYTE   2                 ;PLUS 2 TRANSITIONS
1840
1841 006210 042762 000040 000026      BIC      #DMD,RKMR1(R2)   ;CLEAR DIAG MODE
1842
1843 006216 104423                      TWAT16                      ;WAIT FOR INTERRUPT
1844 006220 104002                      ERROR    2                ;TO SLOW/NOT COMPLETED ERROR
1845
1846 006222 104422                      TCHKWE                      ;CHECK OPERATION WITH ERROR
1847 006224 000000                      .WORD   0
1848 006226 000000                      .WORD   0
1849 006230 000004                      .WORD   UFERR             ;UNIT FIELD ERROR SHOULD SET
1850 006232 104004                      ERROR    4 ; OR 5,6,7    ;REPORT ANY DISCREPENCIES
    
```

.SBTTL \*\*ATTENTION HANDLING BY CONTROLLER

```

*****
;TEST 13      DOUBLE INTERRUPT
;
;      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.
;      MAKE SURE STATUS VALID IS SET.  CHECK THAT SECOND
;      INTERRUPT OCCURS.  AFTER SECOND INTERRUPT
;      CHECK THAT STATUS VALID IS RESET.  ISSUE SELECT
;      AND MAKE SURE STATUS VALID SETS.  CLEAR DRIVE.
;      CHECK THAT DRIVE STATUS CHANGE SETS
;      (BIT 14 OF DRIVE STATUS
;      REGISTER)
*****
    
```

```

1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867 006234 000004                      TST13: SCOPE
1868 006236 012737 000062 001262      MOV      #50.,$TIMES     ;;DO 50. ITERATIONS
1869 006244 104416                      TSSINIT                    ;CLEAR SUBSYSTEM
1870 006246 104003                      ERROR    3                ;BAD INIT ERROR
1871
1872 006250 013737 001626 001610      MOV      DRVNUM,L.CS2     ;LOAD DRIVE NUMBER
1873 006256 012737 000113 001600      MOV      #RECAL,L.CS1    ;LOAD RECAL
    
```

```

1874
1875 006264 104417 TLOADRK ;LOAD RK
1876 006266 104423 TWAT16 ;WAIT FOR 1ST INTERRUPT
1877 006270 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
1878 006272 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
1879 006276 104420 TGETRK ;GET RK REGS
1880 006300 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAL SET
1881 006306 001010 BNE 1$ ;YES-SKIP
1882 006310 012737 053076 001460 MOV #EMSVAL,EM11N
1883 006316 012737 045223 055544 MOV #EMRCAL,DF011A
1884 006324 104011 ERROR 11 ;"SVAL NOT SET RESULT OF RECAL"
1885 006326 000463 BR 50$ ;ABORT TEST
1886
1887 006330 104437 1$: TWAT8S ;WAIT FOR INTERRUPT
1888 006332 000401 BR 2$ ;NO INTERRUPT RETURN
1889 006334 000404 BR 3$ ;INTERRUPT RETURN
1890
1891 006336 012737 052411 001370 2$: MOV #EM50,EM2N ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
1892 006344 104002 ERROR 2
1893
1894 006346 104420 3$: TGETRK ;GET RK REGS
1895 006350 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAC SET NOW
1896 006356 001410 BEQ 4$ ;NO-SKIP
1897 006360 012737 053076 001470 MOV #EMSVAL,EM12N
1898 006366 012737 053323 055544 MOV #EM2INT,DF011A
1899 006374 104012 ERROR 12 ;"SVAL NOT RESET RESULT OF SECOND TEST"
1900 006376 000437 BR 50$
1901
1902 006400 032737 040000 001552 4$: BIT #DSC,T.DS ;TEST DSC SET BY ATTENTION
1903 006406 001010 BNE 5$ ;YES-SKIP
1904 006410 012737 053236 001460 MOV #EMDSC,EM11N
1905 006416 012737 053323 055544 MOV #EM2INT,DF011A
1906 006424 104011 ERROR 11 ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
1907 006426 000423 BR 50$
1908
1909 006430 012737 000101 001600 5$: MOV #SELD, L.CS1 ;LOAD DRIVE SELECT
1910
1911 006436 104417 TLOADRK ;LOAD RK REGS
1912 006440 104423 TWAT16 ;WAIT FOR INTERRUPT
1913 006442 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
1914
1915 006444 104421 TCHKOP ;CHECK FOR ANY ERRORS
1916
1917 006446 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
1918
1919 006450 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAC SET
1920 006456 001007 BNE 50$ ;YES-SKIP
1921 006460 012737 053076 001460 MOV #EMSVAL,EM11N
1922 006466 012737 045134 055544 MOV #EMSELD,DF011A
1923 006474 104011 ERROR 11 ;"SVAL NOT SET RESULT OF DRV SEL.
1924 006476 50$:
1925 ;*****
1926 ;*TEST 14 SINGLE INTERRUPT FROM ATTENTION
1927 ;*
1928 ;* DO A SEEK TO CYLINDER 0. WAIT FOR INTERRUPT FROM
1929 ;* DRIVE ATTENTION. LOWER PRIORITY AGAIN AND MAKE

```

```

1930          ;*      SURE ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.
1931          ;*
1932          ;*****
1933 006476 000004          TST14: SCOPE
1934 006500 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
1935 006506 104416          TSSINIT      ;CLEAR SUBSYSTEM
1936 006510 104003          ERROR      3      ;BAD INIT ERROR
1937
1938 006512 013737 001626 001610      MOV      DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
1939 006520 012737 000117 001600      MOV      #SEEK,L.CS1    ;LOAD SEEK DCYL LEFT AT 0.
1940
1941 006526 104417          TLOADRK      ;LOAD RK REGS
1942 006530 104423          TWAT16      ;WAIT FOR INTERRUPT
1943 006532 104002          ERROR      2      ;TO SLOW/NOT COMPLETED ERROR
1944
1945
1946 006534 104420          TGETRK      ;GET RK REGS
1947
1948 006536 032737 040000 001540      BIT      #DI,T.CS1      ;TEST DI SET
1949 006544 001010          BNE      2$            ;YES-SKIP
1950 006546 012737 053216 001460      MOV      #EMDI,EM11N
1951 006554 012737 053344 055544      MOV      #EMSKSF,DF011A
1952 006562 104011          ERROR      11          ;"DI NOT SET RESULT OF SEEK TO SELF"
1953 006564 000417          BR      50$
1954
1955
1956 006566 012700 000031          2$:      MOV      #25.,RO      ;LOAD AND DECREMENT A COUNT TO
1957 006572 005300          3$:      DEC      RO          ;ZERO. GIVE CONTROLLER A CHANCE TO
1958 006574 001376          BNE      3$            ;INTERRUPT AGAIN.  ERROR IF IT DOES.
1959
1960 006576 022737 000001 001662      CMP      #1,INTSET      ;CHECK ONLY ONE INTERRUPT OCCURRED
1961 006604 001407          BEQ      50$            ;YES-SKIP
1962 006606 012737 053516 001500      MOV      #EMMI,EM13N
1963 006614 012737 053344 055544      MOV      #EMSKSF,DF011A
1964 006622 104013          ERROR      13          ;"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"
1965
1966 006624 104421          50$:     TCHKOP      ;CHECK FOR ANY ERRORS
1967 006626 104004          ERROR      4 ;OR 5,6,7 ;REPORT ALL ERRORS
1968
1969          ;*****
1970          ;*TEST 15      RESET ATTENTIONS WITH UNIBUS INIT
1971          ;*
1972          ;*      DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
1973          ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
1974          ;*
1975          ;*****
1976 006630 000004          TST15: SCOPE
1977 006632 012737 000012 001262      MOV      #10.,$TIMES    ;;DO 10. ITERATIONS
1978 006640 104416          TSSINIT      ;CLEAR SUBSYSTEM
1979 006642 104003          ERROR      3      ;BAD INIT ERROR
1980
1981 006644 013737 001626 001610      MOV      DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
1982 006652 012737 000117 001600      MOV      #SEEK,L.CS1    ;LOAD SEEK (TO SELF-0)
1983
1984 006660 104417          TLOADRK      ;LOAD RK REGS
1985 006662 104423          TWAT16      ;WAIT FOR INTERRUPT
1986 006664 104002          ERROR      2      ;TO SLOW/NOT COMPLETE

```

```

1986
1987 006666 104420          TGETRK          ;GET RK REGS
1988
1989 006670 032737 040000 001540  BIT    #DI,T.CS1    ;TEST DI SET
1990 006676 001010          BNE     1$          ;YES-EXIT
1991 006700 012737 053216 001460  MOV     #EMDI,EM11N
1992 006706 012737 053344 055544  MOV     #EMSKSF,DF011A
1993 006714 104011          ERROR   11          ;"DI NOT SET RESULT OF SEEK TO SELF
1994 006716 000446          BR      50$
1995
1996 006720 005037 001662          1$:  CLR    INTSET    ;CLEAR INTERRUPT COUNTER
1997 006724 000005          RESET   ;DO UNIBUS RESET
1998
1999 006726 005037 001660          CLR    LCLKTK      ;CLEAR TICK COUNTER
2000 006732 004737 032744          JSR    PC,MYTIME   ;CALL TIMER
2001 006736 022737 000012 001660  CMP    #10.,LCLKTK ;COUNT 10 TICKS (MILLISECONDS)?
2002 006744 001372          BNE     5$          ;NO - LOOP
2003
2004 006746 012762 000100 000000  MOV    #IE,RKCS1(R2) ;SET IE FOR ANY STRAY INTERRUPTS
2005 006754 004737 032434          JSR    PC,OPTTST   ;SET UP OPTIONS AGAIN
2006
2007 006760 104423          TWAT16          ;WAIT 16 MS FOR AN INTERRUPT
2008 006762 000410          BR      2$          ;NONE IS EXPECTED SO RETURN SHOULD BE
2009                                ;HERE-BR TO CONTINUE TEST.
2010 006764 012737 053361 001500  MOV    #EMUXIT,EM13N ;INT OCCURRED ON RESET
2011 006772 012737 053423 055544  MOV    #EMRSET,DF011A
2012 007000 104013          ERROR   13          ;"UNEXECUTED INTERRUPT RESULT OF RESET"
2013 007002 000414          BR      50$
2014 007004 104420          2$:  TGETRK          ;GET RK REGS
2015 007006 032737 040000 001540  BIT    #DI,T.CS1    ;TEST DI RESET
2016 007014 001407          BEQ    50$          ;YES-SKIP
2017 007016 012737 053216 001470  MOV    #EMDI,EM12N
2018 007024 012737 053423 055544  MOV    #EMRSET,DF011A
2019 007032 104012          ERROR   12          ;"DI NOT RESET RESULT OF RESET"
2020
2021 007034          50$:
2022
2023          .SBTTL  **ILLEGAL DISK ADDRESS ERROR TESTS
2024
2025          ;*****
2026          ;TEST 16      ILLEGAL DISK ADDRESS (PART 1)
2027          ;
2028          ;      ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
2029          ;      ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
2030          ;      CLEAR CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7,
2031          ;      CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
2032          ;      HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
2033          ;*****
2034          ;*****
2035 007034 000004          T$T16: SCOPE
2036 007036 012737 000062 001262  MOV    #50, $TIMES ;:DO 50. ITERATIONS
2037 007044 012701 000003          MOV    #3,R1      ;PRESET FOR SELECTING TRACK 3
2038
2039 007050 104416          TSSINIT          ;CLEAR SUBSYSTEM
2040 007052 104003          ERROR   3
2041

```

# K06

```

2042 007054 012737 000113 001600      MOV    #RECAL,L.CS1    ;SET UP TO RECAL
2043 007062 013737 001626 001610      MOV    DRVNUM,L.CS2    ;LOAD DRIVE
2044
2045 007070 104417                      TLOADRK                ;LOAD RK REGS
2046
2047 007072 104423                      TWAT16                 ;WAIT FOR 1ST INTERRUPT
2048 007074 104002                      ERROR 2                ;TO SLOW/NOT COMPLETE ERROR
2049
2050 007076 005037 001662              CLR    INTSET          ;CLEAR INTERRUPT FLAG
2051
2052 007102 104437                      TWATBS                 ;WAIT FOR INTERRUPT
2053 007104 104002                      ERROR 2
2054
2055 007106 012737 007114 001110      MOV    #1$, $LPERR    ;SET LOCAL LOOP ON ERROR
2056
2057 007114 104416                      1$: TSSINIT            ;CLEAR SUBSYSTEM
2058 007116 104003                      ERROR 3                ;BAD INIT ERROR
2059
2060 007120 013737 001626 001610      MOV    DRVNUM,L.CS2    ;LOAD DRIVE NUMBER
2061 007126 012737 000117 001600      MOV    #SEEK,L.CS1    ;LOAD SEEK
2062 007134 110137 001607                      MOVB   R1,L.DT        ;LOAD TRACK
2063
2064 007140 104417                      TLOADRK                ;LOAD RK REGS
2065 007142 104423                      TWAT16                 ;WAIT FOR INTERRUPT
2066 007144 104002                      ERROR 2                ;TO SLOW/NOT COMPLETE
2067
2068 007146 032701 000001              BIT    #BIT0,R1        ;TEST IF HEAD ADDRESS HAS BIT 0
2069 007152 001403                      BEQ    2$              ;NO - SKIP
2070 007154 032701 000002              BIT    #BIT1,R1        ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
2071 007160 001007                      BNE    3$              ;YES-GO CHECK BOTH IDAE AND SKI SET
2072
2073 007162 104422                      2$: TCHKWE            ;CHECK OPERATION WITH ERROR
2074 007164 002000                      IDAERR                ;ILLEGAL DISK ADDRESS ERROR
2075 007166 000000                      0
2076 007170 000000                      0
2077 007172 104004                      ERROR 4 ; OR 5,6,7    ;REPORT ALL DISCREPANCIES
2078 007174 104415                      SCOP1                 ;LOCAL LOOP ON ERROR
2079 007176 000406                      BR     4$
2080
2081 007200 104422                      3$: TCHKWE            ;CHECK OPERATION WITH ERROR
2082 007202 002002                      IDAERR!SKIERR        ;ILLEGAL DISK ADDRESS ERROR
2083 007204 000000                      0                    ;SEEK INCOMPLETE
2084 007206 000000                      0
2085 007210 104004                      ERROR 4 ;OR 5,6,7    ;REPORT ANY DISCREPANCIES
2086 007212 104415                      SCOP1                 ;LOCAL LOOP ON ERROR TO 1$
2087
2088 007214 005201                      4$: INC    R1          ;ELSE BUMP TO NEXT ILLEGAL TRACK
2089 007216 022701 000010      CMP    #8.,R1         ;ALL ILLEGAL TRACKS SELECTED?
2090 007222 001334                      BNE    1$             ;NO-LOOP
2091
2092 *****
2093 *TEST 17      ILLEGAL DISK ADDRESS (PART 2)
2094
2095 *          ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
2096 *          ILLEGAL DISK ADDRESS ERROR SETS.  CLEAR CONTROLLER AND DRIVE
2097 *****

```

```

2098 007224 000004          TST17: SCOPE
2099 007226 012737 000062 001262  MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
2100 007234 104416          TSSINIT  ;CLEAR SUBSYSTEM
2101 007236 104003          ERROR    3              ;BAD INIT ERROR
2102
2103 007240 012737 000113 001600  MOV      #RECAL,L.CS1    ;LOAD RECALIBRATE
2104 007246 013737 001626 001610  MOV      DRVNUM,L.CS2    ;LOAD DRIVE
2105
2106 007254 104417          TLOADRK                ;LOAD RK REGS
2107
2108 007256 104423          TWAT16                 ;WAIT FOR 1ST INTERRUPT
2109 007260 104002          ERROR    2              ;TO SLOW/NOT COMPLETE ERROR
2110
2111 007262 005037 001662          CLR      INTSET         ;CLEAR INTERRUPT FLAG
2112
2113 007266 104437          TWAT8S                 ;WAIT FOR INTERRUPT
2114 007270 104002          ERROR    2
2115
2116 007272 012737 007300 001110  MOV      #1$, $LPERR     ;SET LOOP TO BYPASS RECAL
2117
2118 007300 104416          1$: TSSINIT             ;CLEAR SUBSYSTEM
2119 007302 104003          ERROR    3
2120
2121 007304 013737 001626 001610  MOV      DRVNUM, L.CS2   ;LOAD DRIVE NUMBER
2122 007312 012737 000117 001600  MOV      #SEEK,L.CS1    ;LOAD SEEK
2123 007320 012737 001000 001614  MOV      #1000, L.DCYL  ;LOAD ILLEGAL CYLINDER
2124
2125 007326 104417          TLOADRK                ;LOAD RK REGS
2126 007330 104423          TWAT16                 ;WAIT FOR INTERRUPT
2127 007332 104002          ERROR    2              ;TO SLOW/NOT COMPLETE ERROR
2128
2129 007334 104422          TCHKWE                 ;CHECK OPERATION WITH ERROR
2130 007336 002000          .WORD    IDAERR        ;DISK ADDRESS ERROR
2131 007340 000000          .WORD    0
2132 007342 000000          .WORD    0
2133 007344 104004          ERROR    4 ; OR 5,6,7  ;REPORT ANY DISCREPANCIES
2134
2135          .SBTTL  **WRITE HEADER TESTS
2136
2137          ;*****
2138          ;*TEST 20  READ BAD SECTOR INFORMATION
2139          ;*
2140          ;*      ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
2141          ;*      TRACK 2 TO GET THE FACTORY DETECTED BAD
2142          ;*      SECTOR FILE, 26 SECTOR MODE.
2143          ;*
2144          ;*      IF AN ERROR OCCURS, READ SECTOR 2, 4, 6, OR 10(8) UNTIL
2145          ;*      A SUCCESSFUL READ IS DONE. IF NONE READ SUCCESSFULLY
2146          ;*      REMOVE THIS DRIVE FROM TEST. WHEN A READ IS SUCCESSFUL,
2147          ;*      TEST THAT THE PACK IS NOT AN ALIGNMENT PADK AND
2148          ;*      STORE THE ENTRIES FOR LATER USE.
2149          ;*
2150          ;*      REPEAT THIS SERIES OF OPERATIONS FOR FACTORY DETECTED
2151          ;*      BAD SECTORS 24 SECTOR MODE, SOFTWARE DETECTED
2152          ;*      BAD SECTORS 26 SECTOR MODE, AND SOFTWARE DETECTED BAD
2153          ;*      SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR

```

24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED FROM TESTING.

NOTE: THIS TEST IS RUN IN THE FIRST (QUICK VERIFY) PASS ONLY.

\*\*\*\*\*

2154					24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED FROM TESTING.
2155					
2156					
2157					NOTE: THIS TEST IS RUN IN THE FIRST (QUICK VERIFY) PASS ONLY.
2158					
2159					
2160					*****
2161	007346	000004			TST20: SCOPE
2162	007350	012737	000001	001262	MOV #1,\$TIMES ;DO 1 ITERATION
2163	007356	105037	007430		CLRB 2\$ ;CLEAR SECTOR POINTER
2164	007362	005737	001304		TST \$PASS ;TEST IF FIRST PASS
2165	007366	001402			BEQ 15\$ ;NO - SKIP
2166	007370	000137	010134		JMP 28\$ ;ELSE EXIT TEST
2167	007374				
2168	007374	005000			15\$: CLR R0
2169	007376	005005			CLR R5 ;CLEAR R5 FOR BAD SECTOR COUNTING
2170	007400	013703	001640		MOV BSF26P,R3 ;SET POINT IN TO STORE BS 26 SECT FORMAT
2171	007404	012737	007412	001110	MOV #1\$, \$LPERR ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
2172	007412	104416			1\$: TSSINIT ;CLEAR SUBSYSTEM
2173	007414	104003			ERROR 3 ;BAD INIT ERROR
2174					
2175	007416	004437	033324		JSR R4,LRLOAD ;LOAD "L" REGS WITH
2176	007422	000121			RDDATA ; READ DATA
2177	007424	177400			-400 ; WORD COUNT
2178	007426	056446			IBUFF ; BUFFER ADDRESS
2179	007430	000			2\$: .BYTE 0 ; SECTOR ADDRESS
2180	007431	002			.BYTE 2 ; TRACK ADDRESS
2181	007432	000632			632 ; CYLINDER ADDRESS
2182					
2183	007434	104417			TLOADRK ;LOAD "L" REGS INTO RK
2184	007436	104431			TWAT112 ;WAIT FOR INTERRUPT
2185	007440	104002			ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
2186	007442	104421			TCHKOP ;CHECK FOR ANY ERRORS
2187	007444	104004			ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2188	007446	104415			SCOPI ;LOOP TO 1\$ IF SW 9 SET
2189					
2190	007450	105737	001103		TSTB \$ERFLG ;TEST FOR ERROR IN OPERATION
2191	007454	001477			BEQ 7\$ ;NO-SKIP
2192	007456	005700			TST R0 ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
2193	007460	001020			BNE 3\$ ;NO-SKIP
2194	007462	062737	000002	007430	ADD #2,2\$ ;NEXT SECTOR ADDRESS
2195	007470	122737	000012	007430	CMPB #10.,2\$ ;PAST APPLICABLE SECTORS?
2196	007476	001063			BNE 6\$ ;NO-SKIP
2197	007500	012737	052510	001360	MOV #EM51,EMIN
2198	007506	104001			ERROR 1 ;"CANNOT READ BS FILES
2199	007510	043737	001630	001354	25\$: BIC DRVBIT,\$DEVM ;CLEAR DRIVE FROM DRIVE MAP
2200	007516	000137	026174		JMP NEWDRV ;ABORT TEST PASS.
2201					
2202	007522	022700	000001		3\$: CMP #1,R0 ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
2203	007526	001014			BNE 4\$ ;NO-SKIP
2204	007530	062737	000002	007430	ADD #2,2\$ ;NEXT SECTOR ADDRESS
2205	007536	122737	000026	007430	CMPB #22.,2\$ ;PAST APPLICABLE SECTORS?
2206	007544	001040			BNE 6\$ ;NO-SKIP
2207	007546	012737	052510	001360	MOV #EM51,EMIN
2208	007554	104001			ERROR 1 ;"CANNOT READ BS FILES"
2209	007556	000754			BR 25\$

2210									
2211	007560	022700	000002		4\$:	CMP	#2,RO		;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
2212	007564	001014				BNE	5\$		;NO-SKIP
2213	007566	062737	000002	007430		ADD	#2,2\$		;NEXT SECTOR ADDRESS
2214	007574	122737	000013	007430		CMPB	#11.,2\$		;PAST APPLICABLE SECTORS?
2215	007602	001021				BNE	6\$		;NO-SKIP
2216									
2217	007604	012737	052510	001360		MOV	#EM51,EM1N		
2218	007612	104001				ERROR	1		;"CANNOT READ BS FILES"
2219	007614	000735				BR	25\$		
2220									
2221	007616	062737	000002	007430	5\$:	ADD	#2,2\$		;NEXT SECTOR (BS SOFT 24 SECT MODE)
2222	007624	122737	000027	007430		CMPB	#23.,2\$		;PAST APPLICABLE SECTORS?
2223	007632	001005				BNE	6\$		;NO-SKIP
2224	007634	012737	052510	001360		MOV	#EM51,EM1N		
2225	007642	104001				ERROR	1		;"CANNOT READ BS FILES"
2226	007644	000721				BR	25\$		
2227									
2228	007646	105037	001103		6\$:	CLRB	\$ERFLG		;CLEAR ERROR FLAG
2229	007652	000657				BR	1\$		;DO NEXT READ
2230									
2231	007654	005737	056452		7\$:	TST	IBUFF+4		;CHECK FOR ALIGNMENT PACK
2232	007660	001405				BEQ	8\$		;NO-SKIP
2233	007662	012737	052603	001360		MOV	#EM52,EM1N		
2234	007670	104001				ERROR	1		;"ALIGNMENT PACK. DRIVE ABORTING"
2235	007672	000706				BR	25\$		
2236									
2237	007674	012701	056456		8\$:	MOV	#IBUFF+10,R1		;SET TO START OF BAD SECTOR DATA
2238									
2239	007700	022711	177777		9\$:	CMP	#-1,(R1)		;TEST IF WORD ALL ONES (END OF DATA)
2240	007704	001417				BEQ	11\$		;YES-SKIP
2241	007706	012123				MOV	(R1)+,(R3)+		;STORE CYLINDER
2242	007710	012123				MOV	(R1)+,(R3)+		;TRACK AND SECTOR
2243	007712	005205				INC	R5		;BUMP ERROR COUNTER
2244	007714	022705	000025			CMP	#21.,R5		;DOES IT TOTAL 20 FOR THIS FORMAT?
2245	007720	001367				BNE	9\$		;NO-TEST AND MORE NEXT ADDRESS
2246	007722	012737	052661	001360		MOV	#EM53,EM1N		
2247	007730	104001				ERROR	1		;TO MANY BAD SECTORS
2248	007732	043737	001630	001354	10\$:	BIC	DRVBIT,\$DEVM		;CLEAR DRIVE FROM TESTING
2249	007740	000137	026174			JMP	NEWDRV		;ABORT PASS
2250									
2251	007744	005200			11\$:	INC	RO		;BUMP TO NEXT
2252	007746	022700	000001			CMP	#1,RO		;NOW TESTING BS SOFT 26 SECTOR MODE?
2253	007752	001011				BNE	12\$		;NO-SKIP
2254	007754	012723	177777			MOV	#-1,(R3)+		;INSERT END OF FIELD FLAG
2255	007760	010337	001644			MOV	R3,\$SS26P		;SET POINTER TO BAD SECTOR SOFTWARE FIELD
2256	007764	112737	000012	007430		MOVB	#12,2\$		;SET TO FIRST SECTOR THIS MODE
2257	007772	000137	007412			JMP	1\$		;GO READ IT.
2258	007776	022700	000002		12\$:	CMP	#2,RO		;NOW TESTING BS FACT 24 SECTOR MODE?
2259	010002	001014				BNE	13\$		;NO-SKIP
2260	010004	012723	177777			MOV	#-1,(R3)+		;INSERT END OF FIELD FLAG
2261	010010	112737	000001	007430		MOVB	#1,2\$		;SET TO FIRST SECTOR THIS MODE
2262	010016	010537	001646			MOV	R5,\$S26CT		;STORE TOTAL BS COUNT 26 SECTOR MODE
2263	010022	005005				CLR	R5		;CLEAR COUNTER FOR COUNTING 24 SECT BS
2264	010024	013703	001636			MOV	\$SF24P,R3		;SET POINTER FOR STORING BS
2265	010030	000137	007412			JMP	1\$		;GO READ



```

2266
2267 010034 022700 000003 13$: CMP #3,R0 ;NOW TESTING BS SOFT 24 SECTOR MODE
2268 010040 001011 BNE 14$ ;NO-SKIP
2269 010042 012723 177777 MOV #-1,(R3)+ ;INSERT END OF FIELD FLAG
2270 010046 010337 001642 MOV R3,BSS24P ;STORE POINTER TO BSS 24 SECTOR MODE
2271 010052 112737 000013 007430 MOV#B #13,25 ;GET START OF FIELDS BSS 24 SECT MODE
2272 010060 000137 007412 JMP 15 ;GO READ IT
2273
2274 010064 012723 177777 14$: MOV #-1,(R3)+ ;INSERT END OF FIELD FLAG
2275 010070 010537 001650 MOV R5,BS24CT ;STORE COUNT BSS 24 SECTOR MODE
2276
2277
2278 010074 012700 055576 MOV #BS26,R0 ;GET START OF BAD SECTOR BUFFER
2279 010100 012703 056006 MOV #BS24+52.,R3 ;GET END OF BUFFER
2280
2281 010104 022720 000312 27$: CMP #312,(R0)+ ;TEST IF ANY SECTORS BAD IN CYL 312
2282 010110 001403 BEQ 26$ ;YES - GET OUT
2283
2284 010112 020003 CMP R0,R3 ;CHECK IF ALL OF BUFFER TESTED
2285 010114 001373 BNE 27$ ;NO - LOOP
2286 010116 000406 BR 28$ ;EXIT
2287
2288 010120 012737 053542 001360 26$: MOV #DRVABT,EMIN ;"BAD SECTOR IN AREA FOR TEST"
2289 010126 104001 ERROR 1
2290 010130 000137 007510 JMP 25$
2291
2292 010134 28$:
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303 010134 000004
2304 010136 012737 000062 001262
2305 010144 105037 010267
2306 010150 105037 010173
2307 010154 104416
2308 010156 104003
2309
2310 010160 004437 033324 9$: JSR R4,LRLOAD ;LOAD "L" REGS
2311 010164 000127 WRHEAD ;WRITE HEADER
2312 010166 177676 -102 ;WORD CNT FOR 26 SECTOR MODE
2313 010170 060446 OBUFF ;BUFFER
2314 010172 000 .BYTE 0 ;SECTOR 0
2315 010173 000 11$: .BYTE 0 ;TRACK 0
2316 010174 000312 312 ;CYL 0
2317
2318 010176 004437 037526 JSR R4,GENCOM ;GENERATE DATA
2319 010202 000600 000600 ;BUILD HEADERS-NO BAD SECTORS
2320 010204 012737 010214 001110 MOV #111$,SLPERR ;SET LOCAL LOOP ON ERROR
2321 010212 000402 BR 15 ;SKIP INIT

```

```

*****
:TEST 21 FORMAT IN 26 SECTOR FORMAT
:
:
: FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
: FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT
: OCCUR WITH WRITE HEADER ON IN READING DATA
: BUFFER AFTER READ HEADER.
:
*****

```

```

2322 010214          111$:
2323 010214 104416   TSSINIT          ;CLEAR SUBSYSTEM
2324 010216 104003   ERROR 3          ;BAD INIT ERROR
2325 010220 104417   1$: TLOADRK      ;LOAD RK REGS
2326 010222 104431   TWTAT112      ;WAIT FOR INTERRUPT
2327 010224 104002   ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
2328
2329 010226 104421   TCHKOP        ;CHECK FOR ANY ERRORS
2330 010230 104004   ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2331
2332 010232 104415   SCOP1        ;INTERNAL LOOP TO 111$
2333
2334 010234 012737 010274 001110  MOV #112$,SLPERR ;SET LOCAL LOOP ON ERROR
2335 010242 010203      MOV R2,R3       ;BUILD POINTER TO RKDB
2336 010244 062703 000024  ADD #RKDB,R3
2337 010250 012701 056446  MOV #IBUFF,R1   ;SET POINTER TO BUFFER
2338 010254 004437 033324  JSR R4,LRLoad  ;LOAD "L" REGS
2339 010260 000125      RDHEAD         ;READ HEADER
2340 010262 000000      0             ;NO WORDS COUNT
2341 010264 000000      0             ;NO BUFFER
2342 010266 000       .BYTE 0             ;SECTOR 0
2343 010267 000       .BYTE 0             ; TRACK 0
2344 010270 000312      312           ; CYL 312
2345
2346 010272 000402      BR 2$         ;SKIP INIT
2347 010274
2348 010274 104416   112$: TSSINIT        ;CLEAR SUBSYSTEM
2349 010276 104003   ERROR 3        ;BAD INIT ERROR
2350 010300 104417   2$: TLOADRK      ;LOAD RK REGS
2351 010302 104423   TWTAT16      ;WAIT FOR INTERRUPT
2352 010304 104002   ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
2353
2354 010306 104421   TCHKOP        ;CHECK FOR ANY ERRORS
2355 010310 104004   ERROR 4; OR 5,6,7 ;REPORT ALL ERRORS
2356 010312 012700 000003  MOV #3,R0      ;SET COUNT
2357 010316 011321 5$: MOV (R3),(R1)+ ;GET RKDB
2358 010320 104420   TGETRK        ;GET RK REGS
2359 010322 032737 100000 001550 BIT #DLT,T.CS2 ;TEST IF DATA LATE
2360 010330 001410   BEQ 3$        ;NO-SKIP
2361 010332 012737 051303 050732 MOV #EMDLT,EM13
2362 010340 012737 053431 055544 MOV #EMRDB,DF011A
2363 010346 104013   ERROR 13      ;"DATA LATE SET RESULT OF DB READ
2364 010350 104415   SCOP1        ;LOCAL LOOP TO 112$
2365
2366 010352 032737 100000 001540 3$: BIT #CERR,T.CS1 ;TEST IF CONT ERROR SET
2367 010360 001410   BEQ 4$        ;NO-SKIP
2368 010362 012737 053455 001500 MOV #EMCERR,EM13N
2369 010370 012737 053431 055544 MOV #EMRDB,DF011A
2370 010376 104013   ERROR 13      ;"CERR SET RESULT OF READ DB
2371 010400 104415   SCOP1        ;LOCAL LOOP TO 112$
2372 010402 005300 4$: DEC R0         ;DEC COUNT
2373 010404 001344      BNE 5$        ;LOOP IF NOT ZERO
2374 010406 012737 010416 001110 MOV #117$,SLPERR ;SET LOCAL LOOP 117$
2375 010414 000402      BR 7$        ;SKIP INIT
2376 010416
2377 010416 104416   117$: TSSINIT        ;CLEAR SUBSYSTEM

```

```

2378 010420 104003          ERROR 3          ;BAD INIT ERROR
2379 010422 004437 033644 7$: JSR R4,RDSTHD ;GO READ & SEQUENCE HEADERS
2380 010426 104421          TCHKOP          ;CONTROLLER ERROR RETURN
2381 010430 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2382 010432 104013          ERROR 13 ;"DATA LATE SET RESULT OF DATA BUFFER READ"
2383 010434 104002          ERROR 2          ;"OPERATION TOO SLOW" MESSAGE
2384                                     ;OR "HEADER 0 NOT FOUND" MESSAGE
2385
2386 010436 004437 037526 JSR R4,GENCOM
2387 010442 100200          100200          ;COMPARE IBUF & OBUF (HEADERS)
2388 010444 000414          BR 6$           ;GOOD RETURN-NO MISCOMPARES
2389 010446 104015          ERROR 15        ;REPORT 1ST MISCOMPARES
2390
2391 010450 013700 001634 12$: MOV ERRLMT,RO ;GET ERROR LIMIT
2392 010454 005300          DEC RO          ;DECREMENT IT
2393 010456 001407          BEQ 6$         ;EXIT IF ZERO
2394 010460 004437 037526 JSR R4,GENCOM
2395
2396 010464 040000          040000          ;RESUME COMPARE
2397 010466 000403          BR 6$           ;GOOD RETURN-NO MORE ERRORS
2398 010470 104016          ERROR 16        ;REPORT NEXT ERROR LINE
2399 010472 000770          BR 12$         ;LOOP
2400 010474 104415          SCOPI          ;LOCAL ERROR LOOP TO 117$
2401
2402 010476 105737 001607 6$: TSTB L.DT      ;WAS TRACK 1 JUST DONE?
2403 010502 001010          BNE 8$         ;YES-SKIP
2404
2405 010504 112737 000001 010173 MOVB #1,11$    ;CHANGE PARAM TO LOAD "L" WITH
2406 010512 112737 000001 010267 MOVB #1,10$    ;TRACK 2
2407 010520 000137 010160 JMP 9$         ;JUMP TO DO ENTIRE TEST ON TRK 1
2408
2409 010524          8$:

```

.SBTTL \*\*HEADER RECOGNITION TESTS

```

*****
:TEST 22          BAD SECTOR ERROR
:
:   FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
:   SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
:   (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
:   AND ALL OTHER SECTORS GOOD.
:
:   ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
:   SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS.  ISSUE A
:   WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
:   MAKE SURE BAD SECTOR ERROR SET.  ISSUE A WRITE DATA
:   OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2. MAKE
:   SURE NO ERROR SETS.
*****

```

```

2429 010524 000004          TST2: SCOPE
2430 010526 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
2431 010534 104416          TSSINIT        ;CLEAR SUBSYSTEM
2432 010536 104003          ERROR 3        ;BAD INIT ERROR
2433

```



```

2490 010724 104421      6$:  TCHKOP      ;CHECK FOR GOOD OPERATION
2491 010726 104004      ERROR  4 ; OR 5,6,7 ;REPORT ALL ERRORS
2492
2493 010730
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504 010730 000004      7$:  *****
2505 010732 012737 000062 001262  *TEST 23      HEADER VRC ERROR
2506 010740 104416
2507 010742 104003
2508
2509 010744 004437 033324      *
2510 010750 000127      *   FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
2511 010752 177676      *   16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA
2512 010754 060446      *   OF EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE
2513 010756 000      *   SURE HEADER VRC ERROR SETS.  ISSUE A WRITE DATA TO
2514 010757 000      *   A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.
2515 010760 000312      *
2516
2517
2518 010762 004437 037526      *
2519 010766 000600      *
2520 010770 012700 060452      *
2521 010774 012703 000001      *
2522 011000 030310      1$:  SCOPE
2523 011002 001402      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
2524 011004 040310      TSSINIT      ;CLEAR SUBSYSTEM
2525 011006 000401      ERROR  3      ;BAD INIT ERROR
2526 011010 050310      JSR      R4,LRLoad      ;LOAD "L" REGS
2527 011012 062700 000006      WRHEAD      ;WRITE HEADER
2528
2529 011016 006303      -102      ;WORD COUNT
2530 011020 001367      OBUFF      ;BUFF ADD
2531
2532 011022 104417      .BYTE  0      ;SECT
2533 011024 104431      .BYTE  0      ;TRACK
2534 011026 104002      .BYTE  0      ;CYL
2535 011030 104421
2536 011032 104004
2537
2538 011034 012737 011042 001110      JSR      R4,GENCOM      ;BUILD HEADERS NO BSE
2539 011042 104416
2540 011044 104003
2541
2542 011046 004437 033324      MOV      #OBUFF+4,R0      ;GET ADDRESS OF VRC HDRO
2543 011052 000123      MOV      #BIT0,R3      ;SET FOR BIT CHANGE SELECT
2544 011054 177400      1$:  BIT      R3,(R0)      ;CHECK A VRC BIT
2545 011056 060446      BEQ      2$      ;SKIP IF ZERO
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
    
```

```

2546 011060 000          5$: .BYTE 0          ;SECT
2547 011061 000          .BYTE 0          ;TRACK
2548 011062 000312      312          ;CYL
2549
2550 011064 104417      TLOADRK      ;LOAD RK REG
2551 011066 104424      TWAT32      ;WAIT FOR INTERRUPT
2552 011070 104002      ERROR 2     ;TO SLOW/NOT COMPLETE ERROR
2553
2554 011072 022737 000020 011060  CMP #16.,5$ ;WAS THIS WRITE SECTOR 16?
2555 011100 001415      BEQ 6$     ;YES-SKIP
2556
2557 011102 104422      TCHKWE     ;CHECK OPERATION WITH ERROR
2558 011104 000000      0
2559 011106 000040      40        ;HVRC EM EXPECTED
2560 011110 000000      0
2561 011112 104004      ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
2562
2563 011114 104415      SCOPI     ;LOCAL LOOP TO 4$
2564
2565 011116 105237 011060  INCB 5$    ;BUMP SECTOR IN "L" REG
2566 011122 022737 000016 011060  CMP #16,5$ ;IF SECTOR IS 16 OR LESS
2567 011130 003744      BLE 4$    ;LOOP
2568 011132 000402      BR 7$    ;ELSE EXIT
2569 011134 104421      6$: TCHKOP ;CHECK LAST OPERATION NO ERRORS
2570 011136 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2571
2572 011140      7$:
2573 ;*****
2574 ;*TEST 24 BAD SECTOR ERROR AND HVRC ERROR
2575 ;*
2576 ;* FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
2577 ;* BOTH A BAD SECTOR ERROR AND HEADER VRC. ISSUE A WRITE DATA
2578 ;* TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE ONLY HEADER VRC
2579 ;* ERROR SETS.
2580 ;*
2581 ;*****
2582 011140 000004      TST24: SCOPE
2583 011142 012737 000062 001262  MOV #50.,$TIMES ;DO 50. ITERATIONS
2584 011150 104416      TSSINIT    ;CLEAR SUBSYSTEM
2585 011152 104003      ERROR 3    ;BAD INIT ERROR
2586
2587 011154 004437 033324  JSR R4,LRLOAD ;LOAD "L" REG
2588 011160 000127      WRHEAD    ;WRITE HEADER
2589 011162 177676      -102     ;WORD CNT FOR 26 SECTOR MODE
2590 011164 060446      OBUFF    ;BUFF ADD
2591 011166 000          .BYTE 0    ;SECTOR
2592 011167 000          .BYTE 0    ;TRACK
2593 011170 000312      312     ;CYLINDER
2594
2595 011172 004437 037526  JSR R4,GENCOM ;BUILD HEADERS-NO BSE
2596 011176 000600      600
2597
2598 011200 042737 100000 060450  BIC #BIT15,OBUFF+2 ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
2599
2600 011206 104417      TLOADRK    ;LOAD RK REGS
2601 011210 104431      TWAT112   ;WAIT FOR INTERRUPT

```

```

2602 011212 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
2603
2604 011214 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2605 011216 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2606
2607 011220 004437 033324    JSR R4,LRLOAD   ;LOAD "L" REGS
2608 011224 000123          WRDATA          ;WRITE DATA
2609 011226 177400          -400           ;WORD COUNT
2610 011230 060446          OBUFF          ;BUFF ADD
2611 011232 000           .BYTE 0         ;SECTOR
2612 011233 000           .BYTE 0         ;TRACK
2613 011234 000312          312           ;CYLINDER
2614
2615 011236 104417          TLOADRK        ;LOAD RK REGS
2616 011240 104424          TWAT32         ;WAIT FOR INTERRUPT
2617 011242 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
2618
2619 011244 104422          TCHKWE         ;CHECK OPERATION WITH EXPECTED ERR
2620 011246 000000          0
2621 011250 000040          40            ;HVRC ERR EXPECTED
2622 011252 000000          0
2623 011254 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL DISCREPENCIES
2624
2625 *****
2626 *TEST 25 OPERATION INCOMPLETE
2627 *
2628 * FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE
2629 * WRONG FORMAT. ISSUE A WRITE DATA OF 400 TO CYLINDER 0,
2630 * TRACK 0, SECTOR 21. MAKE SURE OPI SET.
2631 *
2632 *****
2633 011256 000004          TST25: SCOPE
2634 011260 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
2635 011266 104416          TSSINIT        ;CLEAR SUBSYSTEM
2636 011270 104003          ERROR 3        ;BAD INIT ERROR
2637
2638 011272 004437 033324    JSR R4,LRLOAD   ;LOAD "L" REGS
2639 011276 000127          WRHEAD         ;WRITE HEADER
2640 011300 177676          -102          ;WORD COUNT FOR 26 SECT MODE
2641 011302 060446          OBUFF          ;BUFF ADD
2642 011304 000           .BYTE 0         ;SECTOR
2643 011305 000           .BYTE 0         ;TRACK
2644 011306 000312          312           ;CYLINDER
2645
2646 011310 004437 037526    JSR R4,GENCOM   ;BUILD HEADERS-NO BSE ERRORS
2647 011314 000600          600
2648
2649 011316 052737 001000 060646 BIS #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
2650 011324 052737 001000 060650 BIS #BIT9,OBUFF+202 ;CORRECT THE VRC
2651
2652 011332 104417          TLOADRK        ;LOAD RK REGS
2653 011334 104431          TWAT112       ;WAIT FOR INTERRUPT
2654 011336 104002          ERROR 2        ;TO SLOW/NOT COMPLETE
2655
2656 011340 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2657 011342 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS

```

```

2658
2659 011344 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
2660 011350 000123 WRDATA ;WRITE DATA
2661 011352 177400 -400 ;400 WORDS
2662 011354 060446 OBUFF ;BUFF ADD
2663 011356 025 .BYTE 25 ;SECTOR 25
2664 011357 000 .BYTE 0 ;TRACK 0
2665 011360 000312 312 ;CYL 312
2666
2667 011362 104417 TLOADRK ;LOAD RK REGS
2668 011364 104425 TWAT48 ;WAIT FOR INTERRUPT
2669 011366 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
2670
2671 011370 104422 TCHKWE ;CHECK OPERATION EXPECTED ERROR
2672 011372 000000 0
2673 011374 000020 20 ;OPI EXPECTED
2674 011376 000000 0
2675 011400 104004 ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
2676
2677 *****
2678 *TEST 26 OPI WITH HVRC ERROR
2679 *
2680 * FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
2681 * ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
2682 * ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
2683 * TRACK 0, SECTOR 17. THAT BOTH OPERATION INCOMPLETE
2684 * AND HEADER VRC SET.
2685 *****
2686 011402 000004 TST26: SCOPE
2687 011404 012737 000062 001262 MOV #50.,$TIMES ;:DO 50. ITERATIONS
2688 011412 104416 TSSINIT ;:CLEAR SUBSYSTEM
2689 011414 104003 ERROR 3 ;:BAD INIT ERROR
2690
2691 011416 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
2692 011422 000127 WRHEAD ;WRITE HEADER
2693 011424 177676 -102 ;WORD COUNT FOR 26 SECT MODE
2694 011426 060446 OBUFF ;BUS ADDRESS
2695 011430 000 .BYTE 0 ;SECTOR
2696 011431 000 .BYTE 0 ;TRACK
2697 011432 000312 312 ;CYLINDER
2698
2699 011434 004437 037526 JSR R4,GENCOM
2700 011440 000600 600 ;BUILD HEADER- NO BSE ERRORS
2701
2702 011442 012700 060602 MOV #OBUFF+134,RO ;GET ADDRESS 2ND WORD HDR 17(8)
2703 011446 052720 001000 BIS #BIT9,(RO)+ ;SET FORMAT 24 SECT PER TRACK
2704 011452 052720 001000 BIS #BIT9,(RO)+ ;SET VRC BIT
2705 011456 062700 000004 ADD #4,RO ;BUMP TO HVRC WORD HDR 20(8)
2706 011462 032710 000001 BIT #BIT0,(RO) ;TEST BIT 0
2707 011466 001403 BEQ 1$ ;RESET-SKIP
2708 011470 042710 000001 BIC #BIT0,(RO) ;CLEAR BIT
2709 011474 000402 BR 2$
2710 011476 052710 000001 1$: BIS #BIT0,(RO) ;SET BIT
2711 ;FORCE OPI AND HVRC ERROR
2712 011502 104417 2$: TLOADRK ;LOAD RK REGS
2713 011504 104431 TWAT112 ;WAIT FOR INTERRUPT

```



```

2714 011506 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
2715
2716 011510 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2717 011512 104004          ERROR 4 ; OR 5,6,7 ;YES-REPORT ALL ERRORS
2718
2719 011514 004437 033324    JSR R4,LRLOAD   ;LOAD "L" REGS
2720 011520 000123          WRDATA         ;WRITE DATA
2721 011522 177400          -400          ;400 WORDS
2722 011524 060446          OBUF          ;BUFF ADDRESS
2723 011526 017            .BYTE 17       ;SECT 17
2724 011527 000            .BYTE 0        ;TRACK 0
2725 011530 000312          312           ;CYLINDER 312
2726
2727 011532 104417          TLOADRK        ;LOAD RK REGS
2728 011534 104425          TWAT48        ;WAIT FOR INTERRUPT
2729 011536 104002          ERROR 2        ;TO SLOW/NOT COMPLETE
2730
2731 011540 104422          TCHKWE        ;CHECK WITH EXPECTED ERROR
2732 011542 000000          0
2733 011544 000060          60            ;HVRC ERR & OPI EXPECTED
2734 011546 000000          0
2735 011550 104004          ERROR 4 ;OR 5,6,7

```

```

*****
*TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
*
*
* FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
* ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
* AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
* END OF THE OPERATION
*
*****

```

```

2745 011552 000004          ST27: SCOPE
2746 011554 012737 000100 001262 MOV #100,$TIMES ;;DO 100 ITERATIONS
2747
2748 011562 104416          TSSINIT        ;CLEAR SUBSYSTEM
2749 011564 104003          ERROR 3        ;BAD INIT ERROR
2750
2751 011566 004437 033324    JSR R4,LRLOAD   ;LOAD "L" REGISTERS
2752 011572 000127          WRHEAD        ;WRITE HEADER
2753 011574 177676          -102         ;WORD COUNT FOR 26 SECTOR MODE
2754 011576 060446          OBUF          ;BUFF ADD
2755 011600 000            .BYTE 0        ;SECTOR
2756 011601 000            .BYTE 0        ;TRACK
2757 011602 000312          312           ;CYLINDER
2758
2759 011604 004437 037526    JSR R4,GENCOM   ;BUILD HEADERS-NO BSE ERRORS
2760 011610 000600          600
2761
2762 011612 012700 060612    MOV #OBUF+144,RO ;ADDRESS OF HEAD 20 HVRC WORD
2763 011616 012701 000002    MOV #BIT1,R1   ;BIT 1 CONSTANT
2764 011622 030110          BIT R1,(R0)    ;TEST BIT 1 SET
2765 011624 001402          BEQ 1$        ;RESET-SKIP
2766 011626 040110          BIC R1,(R0)    ;ELSE CLEAR BIT 1
2767 011630 000401          BR 2$         ;SKIP
2768 011632 050110          BR 1$         ;SKIP
2769
1$: BIS R1,(R0) ;SET BIT 1

```

```

2770 011634 104417          2$: TLOADRK          ;LOAD RK REGS
2771 011636 104431          TWAT112          ;WAIT FOR INTERRUPT
2772 011640 104002          ERROR 2           ;TO SLOW/NOT COMPLETE
2773
2774 011642 104421          TCHKOP          ;CHECK FOR ANY ERROR
2775 011644 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2776
2777 011646 004437 033324   JSR R4,LRLOAD   ;LOAD "L" REGISTER
2778 011652 000123          WRDATA          ;WRITE DATA
2779 011654 177400          -400            ;WORD COUNT
2780 011656 060446          OBUFF           ;BUFF ADD
2781 011660 021             .BYTE 21        ;SECTOR
2782 011661 000             .BYTE 0         ;TRACK
2783 011662 000312          312            ;CYLINDER
2784 011664 104417          TLOADRK          ;LOAD RK REGS
2785 011666 104424          TWAT32          ;WAIT FOR INTERRUPT
2786 011670 104002          ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
2787
2788 011672 104421          TCHKOP          ;CHECK FOR ANY ERROR
2789 011674 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS.
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
    
```

.SBTTL \*\*DATA TRANSFER TESTS

```

*****
:TEST 30 WRITE AND READ ONE SECTOR
:
:
:   FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
:   TO AGREED WITH BAD SECTOR INFORMATION. ISSUE A WRITE DATA
:   OF ONE SECTOR ON CYLINDER 312, TRACK 0. READ IT BACK TO
:   MAKE SURE IT AGREES WITH WHAT IS WRITTEN.
:
*****
    
```

```

2803 011676 000004          TST30: SCOPE
2804 011700 012737 000062 001262  MOV #50.,$TIMES ;:DO 50. ITERATIONS
2805 011706 104416          TSSINIT         ;CLEAR SUBSYSTEM
2806 011710 104003          ERROR 3         ;BAD INIT ERROR
2807
2808 011712 012737 000312 011740  MOV #312,7$     ;PRESET CYL POINTER
2809 011720 105037 011737          CLAB 2$        ;CLEAR TRACK POINTER
2810
2811 011724 004437 033324          1$: JSR R4,LRLOAD ;LOAD "L" REG
2812 011730 000127          WRHEAD         ;WRITE HEADER
2813 011732 177676          -102           ;WORD COUNT FOR 26 SECTOR MODE
2814 011734 060446          OBUFF           ;BUFF ADDRESS
2815 011736 000             .BYTE 0         ;SECTOR
2816 011737 000             .BYTE 0         ;TRACK
2817 011740 000312          7$: 312        ;CYLINDER
2818
2819 011742 004437 037526          JSR R4,GENCOM
2820 011746 001200          1200           ;BUILD HDRS-INCLUDE BAD SECTORS
2821
2822 011750 104417          TLOADRK          ;LOAD RK REGS
2823 011752 104431          TWAT112          ;WAIT FOR INTERRUPT
2824 011754 104002          ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
2825
    
```

```

2826 011756 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2827 011760 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2828
2829 011762 022737 000313 011740  CMP #313,7$ ;TEST IF DONE 313 TK 0
2830 011770 001414          BEQ 3$ ;YES - SKIP
2831 011772 123727 011737 000002  CMPB 2$,#2 ;DID WE JUST FORMAT TRACK 2
2832 012000 001403          BEQ 8$ ;YES-SKIP
2833 012002 105237 011737  INCB 2$ ;BUMP TO NEXT TRACK
2834 012006 000746          BR 1$ ;GO FORMAT NEXT TRACK
2835
2836 012010 105037 011737 8$: CLRB 2$ ;CLEAR TRACK POINTER
2837 012014 005237 011740  INC 7$ ;BUMP CYL TO 313
2838 012020 000741          BR 1$ ;GO FORMAT 313 TK 0
2839
2840 012022 004437 033324 3$: JSR R4,LRLOAD ;LOAD "L" REGS
2841 012026 000123          WRDATA ;WRITE DATA
2842 012030 177400          -400 ;ONE SECTOR WORD COUNT
2843 012032 060446          OBUFF ;BUFF ADDRESS
2844 012034 012          .BYTE 12 ;SECTOR 12
2845 012035 000          .BYTE 0 ;TRACK 0
2846 012036 000312          312 ;CYLINDER 312
2847
2848 012040 004437 037526  JSR R4,GENCOM
2849 012044 000001          1 ;BUILD DATA PATTERN 1
2850 012046 000400          400 ;400 WORDS LONG
2851 012050 012737 012056 001110  MOV #4$, $LPERR ;SET FOR LOCAL LOOP
2852 012056 104417          TLOADRK ;LOAD RK REGS
2853 012060 104431          TWAT112 ;WAIT FOR INTERRUPT
2854 012062 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
2855
2856 012064 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2857 012066 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2858
2859 012070 004437 033324  JSR R4,LRLOAD ;LOAD "L" REGS
2860 012074 000121          RDDATA ;READ DATA
2861 012076 177400          -400 ;400 WORDS
2862 012100 056446          IBUFF ;BUFF ADD
2863 012102 012          .BYTE 12 ;SECTOR 12
2864 012103 000          .BYTE 0 ;TRACK 0
2865 012104 000312          312 ;CYL 312
2866
2867 012106 104417          TLOADRK          ;LOAD RK
2868 012110 104424          TWAT32 ;WAIT FOR INTERRUPT
2869 012112 104002          ERROR 2 ;TO SLOW/NOT COMPLETE
2870
2871 012114 104421          TCHKOP          ;CHECK FOR ANY ERRORS
2872 012116 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
2873
2874 012120 004437 037526  JSR R4,GENCOM
2875 012124 100001          100001 ;GO COMPARE DATA TO PATTERN 1
2876 012126 000400          400 ;400 WORDS LONG
2877 012130 000413          BR 6$ ;GOOD RETURN-NO DATA ERRORS
2878 012132 104015          ERROR 15 ;ERROR RETURN
2879
2880 012134 013700 001634  MOV ERRLMT,RO ;GET ERROR LIMIT
2881 012140 005300          DEC RO ;DEC LIMIT

```

```

2882 012142 001406          BEQ      6$          ;EXIT IF 0
2883 012144 004437 037526   JSR      R4,GENCOM
2884 012150 040000          040000          ;CONTINUE COMPARE
2885 012152 000402          BR       6$          ;EXIT IF NO MORE ERRORS
2886 012154 104016          ERROR   16          ;ELSE REPORT MISCOMPARE
2887 012156 000770          BR       5$          ;LOOP
2888 012160
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937

```

6\$:  
:\*\*\*\*\*  
:TEST 31 WRITE DATA ADDRESS GREATER THAN 32K  
:  
:ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.  
:MAKE SURE CORRECT DATA IS ON DISK.  
:  
:NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K  
:OF MEMORY IS PRESENT.  
:\*\*\*\*\*  
TST31: SCOPE  
MOV #50,\$TIMES ;:DO 50. ITERATIONS  
CMPB \$MAMS1,#1 ;:TEST IF >32K MEM  
BGE 2\$ ;:YES-SKIP  
BIT #MEMSZB,OPTFLG ;:TEST IF REPORT ALREADY MADE  
BNE 1\$ ;:YES -SKIP  
TYPE ,OPR011 ;:"INSUFFICIENT MEMORY DATA TRANSFER WITH  
TYPE ,OPR012 ;:ADDRESS >32K  
BIS #MEMSZB,OPTFLG ;:SET FLAG  
TYPE ,OPR015 ;:BYPASSED"  
BR 4\$ ;:EXIT  
MOV #5\$,\$LPERR ;:SET LOCAL LOOP ON ERROR ADDRESS  
1\$:  
2\$:  
5\$:  
TSSINIT ;:CLEAR SUBSYSTEM  
ERROR 3 ;:BAD INIT ERROR  
JSR R4,LRLOAD ;:LOAD "L" REGS  
WRDATA ;:WRITE DATA  
-400 ;:400 WORDS  
177770 ;:BUS ADDRESS IN 32K -10 BYTES  
.BYTE 16 ;:SECTOR 16  
.BYTE 0 ;:TRACK 0  
312 ;:CYLINDER 312  
JSR R4,GENCOM ;:GENERATE DATA  
10010 ;:PATTERN 10, MEM. MANAGEMENT FOR DEST.  
1777 ;:RELOCATION ARGUMENT  
400 ;:400 WORDS  
TLOADRK ;:LOAD RK REGS  
TWAT32 ;:WAIT FOR INTERRUPT  
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR  
TCHKOP ;:CHECK OPERATION FOR ANY ERRORS  
ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS  
SCOPI ;:LOCAL LOOP ON ERROR TO 5\$  
JSR R4,GENCOM ;:CLEAR Ibuff TO 1'S.

```

2938 012316 002007          2007
2939 012320 001000          1000
2940
2941 012322 004437 033324   JSR      R4,LRLOAD      ;LOAD "L" REGS
2942 012326 000121          RDDATA      ;RDDATA
2943 012330 177400          -400        ;-400 WORDS
2944 012332 056446          Ibuff      ;IBUFF IS BUFF ADDRESS
2945 012334      016        .BYTE      16        ;SECTOR 16
2946 012335      000        .BYTE      0         ;TRACK 0
2947 012336 000312          312        ;CYLINDER 312
2948 012340 104417          TLOADRK    ;LOAD RK REGS
2949 012342 104424          TWAT32     ;WAIT FOR INTERRUPT
2950 012344 104002          ERROR     2         ;TO SLOW/NOT COMPLETE ERROR
2951 012346 104421          TCHKOP    ;CHECK OPERATION FOR ANY ERRORS
2952 012350 104004          ERROR     4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
2953 012352 004437 037526   JSR      R4,GENCOM    ;COMPARE
2954 012356 110000          110000     ;MEMORY MANAGEMENT FOR DESTINATION
2955 012360 001777          1777      ;RELOCATION ARGUMENT
2956 012362 000400          400       ;400 WORDS
2957 012364 000413          BR        4$        ;NO ERROR-SKIP
2958 012366 104015          ERROR     15       ;REPORT FIRST MISCOMPARE
2959 012370 013700 001634   MOV      ERRLMT,R0   ;GET ERROR LIMIT
2960 012374 005300          DEC      R0        ;DECREMENT COUNT
2961 012376 001406          BEQ      65$       ;IF ZERO - EXIT
2962 012400 004437 037526   JSR      R4,GENCOM    ;CONTINUE DATA COMPARE
2963 012404 050000          50000
2964 012406 000402          BR        65$       ;NO MORE ERRORS - EXIT
2965 012410 104016          ERROR     16       ;REPORT NEXT ERROR
2966 012412 000770          BR        64$       ;LOOP
2967 012414
2968
2969 012414
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980 012414 000004          TST32: SCOPE
2981 012416 012737 000062 001262  MOV      #50,$TIMES   ;DO 50. ITERATIONS
2982 012424 123727 001326 000001  CMPB    $MAMS1,#1    ;CHECK IF >32K MEMORY
2983 012432 002001          BGE      2$        ;YES-SKIP
2984
2985 012434 000462          1$: BR        5$        ;EXIT
2986
2987 012436 012737 012444 001110 2$: MOV      #3$,$LPERR ;SET LOCAL ERROR LOOP
2988
2989 012444          3$:
2990 012444 104416          TSSINIT    ;CLEAR SUBSYSTEM
2991 012446 104003          ERROR     3         ;BAD INIT ERROR
2992 012450 004437 033324   JSR      R4,LRLOAD    ;LOAD "L" REGS
2993 012454 000123          WRDATA    ;WRDATA

```

```

2994 012456 177400 -400 ; -400 WORDS
2995 012460 060446 OBUFF ; OBUFF IS BUFF ADDRESS
2996 012462 017 .BYTE 17 ; SECTOR 17
2997 012463 000 .BYTE 0 ; TRACK 0
2998 012464 000312 312 ; CYLINDER 312
2999 012466 004437 037526 JSR R4,GENCOM ; GENERATE DATA IN OBUFF
3000 012472 000011 11 ; PATTERN 11
3001 012474 000400 400 ; 400 WORDS
3002
3003 012476 104417 TLOADRK ; LOAD RK REGS
3004 012500 104424 TWAT32 ; WAIT FOR INTERRUPT
3005 012502 104002 ERROR 2 ; TO SLOW/NOT COMPLETE ERROR
3006
3007 012504 104421 TCHKOP ; CHECK OPERATION FOR ANY ERRORS
3008 012506 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3009 012510 004437 033324 JSR R4,LLOAD ;LOAD "L" REG
3010 012514 000121 RDDATA ;READ DATA
3011 012516 177400 -400 ;400 WORDS
3012 012520 177770 177770 ;ACROSS 32K BOUNDARY
3013 012522 017 .BYTE 17 ;SECTOR 17
3014 012523 000 .BYTE 0 ;TRACK 0
3015 012524 000312 312 ;CYL 312
3016
3017 012526 104417 TLOADRK ;LOAD RK REGS
3018 012530 104424 TWAT32 ;WAIT FOR INTERRUPT
3019 012532 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3020 012534 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
3021 012536 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3022 012540 004437 037526 JSR R4,GENCOM ;COMPARE DATA
3023 012544 120000 120000 ;MEMORY MANAGEMENT WITH SOURCE
3024 012546 001777 1777 ;RELOCATION ARGUMENT
3025 012550 000400 400 ;COMPARE 400 WORDS
3026 012552 000413 BR 5$ ;NO MISCOMPARE-EXIT
3027 012554 104015 ERROR 15 ;REPORT FIRST MISCOMPARE
3028 012556 013700 001634 MOV ERRLMT,R0 ;GET ERROR LIMIT
3029 012562 005300 64$: DEC R0 ;DECREMENT COUNT
3030 012564 001406 BEQ 65$ ;IF ZERO - EXIT
3031 012566 004437 037526 JSR R4,GENCOM ;CONTINUE DATA COMPARE
3032 012572 060000 60000
3033 012574 000402 BR 65$ ;NO MORE ERRORS - EXIT
3034 012576 104016 ERROR 16 ;REPORT NEXT ERROR
3035 012600 000770 BR 64$ ;LOOP
3036 012602
3037 012602
3038
3039 *****
3040 *TEST 33 WRITE DATA ADDRESS GREATER THAN 64K
3041 *
3042 * ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.
3043 * MAKE SURE CORRECT DATA IS ON DISK.
3044 *
3045 * NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
3046 * OF MEMORY IS PRESENT.
3047 *
3048 *****
3049 TST33: SCOPE
MOV #50.,$TIMES ;;DO 50. ITERATIONS

```

3050	012612	123727	001326	000002		CMPB	\$MAMS1,#2	:CHECK IF >64K MEMORY
3051	012620	002016				BGE	2\$	:YES-SKIP
3052	012622	032737	000002	001656	6\$:	BIT	#MEMSZB,OPTFLG	:TEST IF REPORT FLAG SET
3053	012630	001011				BNE	1\$	:NO-SKIP
3054								
3055	012632	104401	046444			TYPE	,OPR011	:INSUFFICIENT MEMORY-DATA XFER WITH
3056	012636	104401	046613			TYPE	,OPR013	:ADDRESS >64K
3057	012642	104401	046623			TYPE	,OPR015	:BYPASSED"
3058	012646	052737	000002	001656		BIS	#MEMSZB,OPTFLG	:SET FLAG
3059	012654	000466			1\$:	BR	5\$	
3060								
3061	012656	012737	012664	001110	2\$:	MOV	#3\$, \$LPERR	:SET LOCAL LOOP ON ERROR
3062								
3063	012664				3\$:			
3064	012664	104416				TSSINIT		:CLEAR SUBSYSTEM
3065	012666	104003				ERROR	3	:BAD INIT ERROR
3066	012670	004437	037526			JSR	R4,GENCOM	:GENERATE DATA, PATTERN 11
3067	012674	010011				10011		:MEM MANAGEMENT ON DESTINATION
3068	012676	003777				3777		:RELOCATION ARGUMENT
3069	012700	000400				400		:400 WORDS
3070								
3071	012702	004437	033324			JSR	R4,LRLOAD	:LOAD "L" REGS
3072	012706	000523				WRDATA:BA16		:WRITE DATA AND SET BA16
3073	012710	177400				-400		:400 WORDS
3074	012712	177770				177770		:ACROSS 64K BOUNDARY
3075	012714	020				.BYTE	20	:SECTOR 20
3076	012715	000				.BYTE	0	:TRACK 0
3077	012716	000312				312		:CYLINDER 312
3078								
3079	012720	104417				TLOADRK		:LOAD RK REGS
3080	012722	104424				TWAT32		:WAIT FOR INTERRUPT
3081	012724	104002				ERROR	2	:TO SLOW/NOT COMPLETE ERROR
3082								
3083	012726	104421				TCHKOP		:CHECK OPERATION FOR ANY ERRORS
3084	012730	104004				ERROR	4 ;OR 5, 6, 7, 10	:REPORT ALL ERRORS
3085	012732	004437	037526			JSR	R4,GENCOM	:CLEAR Ibuff TO 1'S
3086	012736	002007				2007		
3087	012740	004437	033324			JSR	R4,LRLOAD	:LOAD "L" REGS
3088	012744	000121				RDDATA		:RDDATA
3089	012746	177400				-400		:400 WORDS
3090	012750	056446				IBUFF		:IBUFF IS BUFF ADDRESS
3091	012752	020				.BYTE	20	:SECTOR 20
3092	012753	000				.BYTE	0	:TRACK 0
3093	012754	000312				312		:CYLINDER 312
3094	012756	104417				TLOADRK		:LOAD RK REGS
3095	012760	104424				TWAT32		:WAIT FOR INTERRUPT
3096	012762	104002				ERROR	2	:TO SLOW/NOT COMPLETE ERROR
3097								
3098	012764	104421				TCHKOP		:CHECK OPERATION FOR ANY ERRORS
3099	012766	104004				ERROR	4 ;OR 5, 6, 7, 10	:REPORT ALL ERRORS
3100	012770	004437	037526			JSR	R4,GENCOM	:CHECK DATA
3101	012774	110000				110000		:MEMORY MANAGEMENT WITH DESTINATION
3102	012776	003777				3777		:RELOCATION ARGUMENT
3103	013000	000400				400		:400 WORDS
3104	013002	000413				BR	5\$	:NO MISCOMPARES-SKIP
3105	013004	104015				ERROR	15	:REPORT FIRST ERROR

```

3106
3107 013006 013700 001634      MOV      ERRLMT,R0      ;GET ERROR LIMIT
3108 013012 005300      DEC      R0             ;DECREMENT COUNT
3109 013014 001406      BEQ      65$           ;IF ZERO - EXIT
3110 013016 004437 037526      JSR      R4,GENCOM     ;CONTINUE DATA COMPARE
3111 013022 050000      50000
3112 013024 000402      BR       65$           ;NO MORE ERRORS - EXIT
3113 013026 104016      ERROR   16            ;REPORT NEXT ERROR
3114 013030 000770      BR       64$           ;LOOP
3115 013032
3116 013032
3117
3118 *****
3119 *TEST 34      READ DATA ADDRESS GREATER THAN 64K
3120 *
3121 *      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
3122 *      CHECK MEMORY FOR CORRECT TRANSFER.
3123 *
3124 *      NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
3125 *      OF MEMORY IS PRESENT.
3126 *****
3127 013032 000004      TST34:  SCOPE
3128 013034 012737 000062 001262      MOV      #50,$TIMES    ;DO 50. ITERATIONS
3129 013042 123727 001326 000002      CMPB    $MAMS1,#2     ;CHECK IF >64K MEMORY
3130 013050 002001      BGE     2$            ;YES-SKIP
3131 013052 000462      BR      5$            ;EXIT
3132
3133 013054 012737 000032 001110      1$:     MOV      #32,$LPERR ;SET LOCAL LOOP ON ERROR
3134
3135 013062
3136 013062 104416      3$:     TSSINIT      ;CLEAR SUBSYSTEM
3137 013064 104003      ERROR   3            ;BAD INIT ERROR
3138 013066 004437 033324      JSR      R4,LRLOAD    ;LOAD "L" REGS
3139 013072 000123      WRDATA  ;WRDATA
3140 013074 177400      -400    ;-400 WORDS
3141 013076 060446      OBUFF   ;OBUFF IS BUFF ADDRESS
3142 013100 021      .BYTE  21            ;SECTOR 21
3143 013101 000      .BYTE  0             ;TRACK 0
3144 013102 000312      312     ;CYLINDER 312
3145 013104 004437 037526      JSR      R4,GENCOM     ;GENERATE DATA
3146 013110 000012      12      ;PATTERN 12
3147 013112 000400      400     ;400 WORDS
3148
3149 013114 104417      TLOADRK ;LOAD RK REGS
3150 013116 104424      TWAT32  ;WAIT FOR INTERRUPT
3151 013120 104002      ERROR   2            ;TO SLOW/NOT COMPLETE ERROR
3152
3153 013122 104421      TCHKOP  ;CHECK OPERATION FOR ANY ERRORS
3154 013124 104004      ERROR   4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3155 013126 004437 033324      JSR      R4,LRLOAD    ;LOAD "L" REGS
3156 013132 000521      RDDATA!BA16 ;READ DATA AND SET BA16
3157 013134 177400      -400    ;400 WORDS
3158 013136 177770      177770 ;ACROSS 64K BOUNDARY
3159 013140 021      .BYTE  21            ;FROM SECTOR 21
3160 013141 000      .BYTE  0             ;TRACK 0
3161 013142 000312      312     ;CYLINDER 312

```



```

3162
3163 013144 104417          TLOADRK          ;LOAD RK REGS
3164 013146 104424          TWAT32          ;WAIT FOR INTERRUPT
3165 013150 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
3166
3167 013152 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
3168 013154 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3169 013156 004437 037526  JSR R4,GENCOM  ;COMPARE DATA
3170 013162 120000          120000         ;MEM MANAGEMENT WITH SOURCE
3171 013164 003777          3777           ;RELOCATION ARGUMENT
3172 013166 000400          400           ;400 WORDS
3173 013170 000413          BR 5$         ;NO MISCOMPARES-SKIP
3174 013172 104015          ERROR 15      ;REPORT FIRST ERROR
3175
3176 013174 013700 001634  MOV ERRLMT,R0  ;GET ERROR LIMIT
3177 013200 005300          64$: DEC R0    ;DECREMENT COUNT
3178 013202 001406          BEQ 65$      ;IF ZERO - EXIT
3179 013204 004437 037526  JSR R4,GENCOM  ;CONTINUE DATA COMPARE
3180 013210 060000          60000
3181 013212 000402          BR 65$      ;NO MORE ERRORS - EXIT
3182 013214 104016          ERROR 16     ;REPORT NEXT ERROR
3183 013216 000770          BR 64$      ;LOOP
3184 013220          65$:
3185
3186 013220          5$:
3187 *****
3188 ;*TEST 35 WRITE DATA ADDRESS GREATER THAN 96K
3189 ;*
3190 ;* ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
3191 ;* MAKE SURE CORRECT DATA IS ON DISK.
3192 ;*
3193 ;* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
3194 ;* OF MEMORY IS PRESENT.
3195 ;*
3196 *****
3197 013220 000004          TST35: SCOPE
3198 013222 012737 000062 001262  MOV #50,$TIMES ;DO 50. ITERATIONS
3199 013230 123727 001326 000003  CMPB $MAMS1,#3 ;CHECK IF >96K MEMORY
3200 013236 002016          BGE 3$       ;YES-SKIP
3201 013240 032737 000002 001656 1$: BIT #MEMSZB,OPTFLG ;TEST IF REPORT FLAG SET
3202 013246 001011          BNE 2$       ;NO-SKIP
3203
3204 013250 104401 046444          TYPE ,OPR011 ;"INSUFFICIENT MEMORY-DATA TRANSFET WITH
3205 013254 104401 046617          TYPE ,OPR014 ;ADDRESS >96K BYPASSED"
3206 013260 104401 046623          TYPE ,OPR015
3207 013264 052737 000002 001656 2$: BIS #MEMSZB,OPTFLG ;SET REPORT FLAG
3208 013272 000464          BR 6$
3209
3210 013274 012737 013302 001110 3$: MOV #4$,SLPERR ;SET LOCAL LOOP ON ERROR
3211
3212 013302          4$:
3213 013302 104416          TSSINIT      ;CLEAR SUBSYSTEM
3214 013304 104003          ERROR 3      ;BAD INIT ERROR
3215 013306 004437 033324  JSR R4,LRLOAD ;LOAD "L" REG
3216 013312 001123          WRDATA!BA17 ;WRITE DATA AND BA17
3217 013314 177400          -400        ;400 WORDS FROM

```

```

3218 013316 177770          177770          ;ACROSS 96K BOUNDARY
3219 013320          022          ;TO SECTOR 22
3220 013321          000          ;TRACK 0
3221 013322 000312          312          ;CYL 312
3222 013324 004437 037526 JSR          R4,GENCOM ;GENERATE DATA
3223 013330 010013          10013         ;PATTERN 13 MEM MAN WITH DEST.
3224 013332 005777          5777          ;RELOCATION ARGUMENT
3225 013334 000400          400          ;400 WORDS
3226
3227 013336 104417          TLOADRK       ;LOAD RK REGS
3228 013340 104424          TWAT32        ;WAIT FOR INTERRUPT
3229 013342 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
3230
3231 013344 104421          TCHKOP       ;CHECK OPERATION FOR ANY ERRORS
3232 013346 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3233 013350 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3234
3235 013352 004437 033324 JSR          R4,LRLOAD ;LOAD "L" REGS
3236 013356 000121          RDDATA       ;RDDATA
3237 013360 177400          -400         ;-400 WORDS
3238 013362 056446          Ibuff        ;IBUFF IS BUFF ADDRESS
3239 013364          022          ;SECTOR 22
3240 013365          000          ;TRACK 0
3241 013366 000312          312          ;CYLINDER 312
3242
3243 013370 104417          TLOADRK       ;LOAD RK REGS
3244 013372 104424          TWAT32        ;WAIT FOR INTERRUPT
3245 013374 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
3246
3247 013376 104421          TCHKOP       ;CHECK OPERATION FOR ANY ERRORS
3248 013400 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3249
3250 013402 004437 037526 JSR          R4,GENCOM ;COMPARE DATA
3251 013406 110000          110000       ;MEM MANAGEMENT WITH DESTINATION
3252 013410 005777          5777          ;RELOCATION ARGUMENT
3253 013412 000400          400          ;400 WORDS
3254 013414 000413          BR           65 ;NO MISCOMPARES-BRANCH
3255 013416 104015          ERROR 15     ;REPORT 1ST ERROR
3256
3257 013420 013700 001634 MOV          ERRLMT,R0 ;GET ERROR LIMIT
3258 013424 005300          64$: DEC      R0      ;DECREMENT COUNT
3259 013426 001406          BEQ         65$     ;IF ZERO - EXIT
3260 013430 004437 037526 JSR          R4,GENCOM ;CONTINUE DATA COMPARE
3261 013434 050000          50000
3262 013436 000402          BR           65$     ;NO MORE ERRORS - EXIT
3263 013440 104016          ERROR 16     ;REPORT NEXT ERROR
3264 013442 000770          BR           64$     ;LOOP
3265
3266
3267 013444          65$:
3268          *****
3269          ;*TEST 36 READ DATA ADDRESS GREATER THAN 96K
3270          ;*
3271          ;* ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
3272          ;* CHECK MEMORY FOR CORRECT TRANSFER.
3273          ;*
    
```

```

3274      : *      NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
3275      : *      OF MEMORY IS PRESENT.
3276      : *
3277      : *****
3278 013444 000004      TST36: SCOPE
3279 013446 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3280 013454 123727 001326 000003      CMPB     $MAMS1,#3      ;;CHECK IF >96K MEMORY
3281 013462 002001      BGE      3$             ;;YES-SKIP
3282 013464 000462      2$: BR      6$
3283
3284 013466 012737 013474 001110 3$: MOV      #4$,SLPERR      ;SET LOCAL LOOP ON ERROR
3285
3286 013474      4$:
3287 013474 104416      TSSINIT      ;CLEAR SUBSYSTEM
3288 013476 104003      ERROR 3      ;BAD INIT ERROR
3289
3290 013500 004437 033324      JSR      R4,LRLoad      ;LOAD "L" REGS
3291 013504 000123      WRDATA      ;WRDATA
3292 013506 177400      -400        ;-400 WORDS
3293 013510 060446      OBUFF       ;OBUFF IS BUFF ADDRESS
3294 013512 005         .BYTE 5      ;SECTOR 5
3295 013513 000         .BYTE 0      ;TRACK 0
3296 013514 000312      312        ;CYLINDER 312
3297 013516 004437 037526      JSR      R4,GENCOM      ;GENERATE DATA
3298 013522 000014      14         ;PATTERN 14
3299 013524 000400      400        ;400 WORDS
3300
3301 013526 104417      TLOADRK     ;LOAD RK REGS
3302 013530 104424      TWAT32     ;WAIT FOR INTERRUPT
3303 013532 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
3304
3305 013534 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
3306 013536 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3307 013540 004437 033324      JSR      R4,LRLoad      ;LOAD "L" REGS
3308 013544 001121      RDDATA!BA17 ;READ DATA WITH BA17 SET
3309 013546 177400      -400        ;400 WORDS
3310 013550 177770      177770     ;ACROSS 96K BOUNDARY
3311 013552 005         .BYTE 5      ;FROM SECTOR 5
3312 013553 000         .BYTE 0      ;TRACK 0
3313 013554 000312      312        ;CYL 312
3314
3315 013556 104417      TLOADRK     ;LOAD RK REGS
3316 013560 104424      TWAT32     ;WAIT FOR INTERRUPT
3317 013562 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
3318
3319 013564 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
3320 013566 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3321 013570 004437 037526      JSR      R4,GENCOM      ;COMPARE DATA
3322 013574 120000      120000     ;MEM MANAGEMENT WITH SOURCE
3323 013576 005777      5777      ;RELOCATION ARGUMENT
3324 013600 000400      400        ;400 WORDS
3325 013602 000413      BR      6$      ;NO MISCOMPARES-SKIP
3326 013604 104015      ERROR 15     ;REPORT FIRST ERROR
3327
3328 013606 013700 001634      64$: MOV     ERR_LMT,RO    ;GET ERROR LIMIT
3329 013612 005300      DEC      RO      ;DECREMENT COUNT

```

```

3330 013614 001406          BEQ      65$          ;IF ZERO - EXIT
3331 013616 004437 037526  JSR      R4,GENCOM  ;CONTINUE DATA COMPARE
3332 013622 060000          60000
3333 013624 000402          BR       65$          ;NO MORE ERRORS - EXIT
3334 013626 104016          ERROR   16          ;REPORT NEXT ERROR
3335 013630 000770          BR       64$          ;LOOP
3336 013632
3337
3338 013632
3339
3340
3341
3342
3343
3344
3345
3346
3347 013632 000004          TST37: SCOPE
3348 013634 012737 000062 001262  MOV      #50.,$TIMES ;:DO 50. ITERATIONS
3349 013642 104416          TSSINIT ;:CLEAR SUBSYSTEM
3350 013644 104003          ERROR   3          ;:BAD INIT ERROR
3351
3352 013646 004437 033324  JSR      R4,LRLOAD  ;:LOAD "L" REG
3353 013652 000123          WRDATA  ;:WRITE DATA
3354 013654 177675          -103    ;:WORD COUNT PARTIAL SECTOR
3355 013656 060446          OBUFF   ;:BUFF ADDRESS
3356 013660 007          .BYTE   7          ;:SECTOR 7
3357 013661 000          .BYTE   0          ;:TRACK 0
3358 013662 000312          312    ;:CYLINDER 312
3359
3360 013664 004437 037526  JSR      R4,GENCOM  ;:GENERATE DATA
3361 013670 000003          3       ;:PATTERN 3
3362 013672 000400          400    ;:400 WORDS
3363
3364 013674 104417          TLOADRK ;:LOAD RK REGS
3365 013676 104424          TWAT32  ;:WAIT FOR INTERRUPT
3366 013700 104002          ERROR   2          ;:TO SLOW/NOT COMPLETE ERROR
3367
3368 013702 104421          TCHKOP  ;:CHECK FOR ANY ERROR
3369 013704 104004          ERROR   4 ; OR 5,6,7 ;:REPORT ALL ERROR
3370
3371 013706 004437 033324  JSR      R4,LRLOAD  ;:LOAD "L" REGS
3372
3373 013712 000121          RDDATA  ;:READ DATA
3374 013714 177400          -400   ;:ONE FULL SECTOR
3375 013716 056446          IBUFF   ;:BUFF ADDRESS
3376 013720 007          .BYTE   7          ;:SECTOR 7
3377 013721 000          .BYTE   0          ;:TRACK 0
3378 013722 000312          312    ;:CYLINDER 312
3379
3380 013724 004437 037526  JSR      R4,GENCOM  ;:CLEAR Ibuff TO ALL ONES
3381 013730 002007          2007
3382 013732 000400          400
3383
3384 013734 104417          TLOADRK ;:LOAD RK REGS
3385 013736 104424          TWAT32  ;:WAIT FOR INTERRUPT

```

```

3386 013740 104002          ERROR 2          ;TO SLOW/NOT COMPLETE
3387
3388 013742 104421          TCHKOP          ;CHECK FOR ANY ERRORS
3389 013744 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
3390
3391 013746 012701 060654    MOV #OBUFF+206,R1 ;CLEAR THE LAST 205 WORDS
3392 013752 012700 000275    MOV #275,R0      ;OF THE OUTPUT BUFFER TO ZERO
3393 013756 005021          1$: CLR (R1)+     ;TO VERIFY THE PARTIAL SECTOR
3394 013760 005300          DEC R0           ;WRITE 0 FILLED THE SECTOR
3395 013762 001375          BNE 1$
3396 013764 004437 037526    JSR R4,GENCOM
3397 013770 100000          100000         ;COMPARE OBUFF & IBUFF.
3398 013772 000400          400            ;ALL 400 WORDS
3399 013774 000413          BR 3$          ;NO ERRORS-EXIT
3400 013776 104015          ERROR 15       ;REPORT FIRST COMPARE ERROR
3401
3402 014000 013700 001634    2$: MOV ERLMT,R0  ;GET ERROR LIMIT
3403 014004 005300          DEC R0         ;DECREMENT IT
3404 014006 001406          BEQ 3$        ;IF ZERO-EXIT
3405 014010 004437 037526    JSR R4,GENCOM
3406 014014 040000          40000         ;CONTINUE COMPARE
3407 014016 000402          BR 3$         ;NO MORE ERRORS-EXIT
3408 014020 104016          ERROR 16      ;REPORT NEXT COMPARE ERROR
3409 014022 000770          BR 2$        ;LOOP
3410
3411 014024          3$:
3412 *****
3413 *TEST 40 PARTIAL SECTOR READ DATA
3414 *
3415 * WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A
3416 * KNOWN CONFIGURATION. ISSUE A READ DATA OF
3417 * 103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0.
3418 * MAKE SURE ONLY 103 WORDS GET TRANSFERRED
3419 * TO MEMORY.
3420 *
3421 *****
3422 014024 000004          TST40: SCOPE
3423 014026 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
3424 014034 104416          TSSINIT       ;CLEAR SUBSYSTEM
3425 014036 104003          ERROR 3       ;BAD INIT ERROR
3426
3427 014040 004437 033324    JSR R4,LRLOAD ;LOAD "L" REGS
3428 014044 000123          WRDATA       ;WRDATA
3429 014046 177400          -400         ;-400 WORDS
3430 014050 060446          OBUFF        ;OBUFF IS BUFF ADDRESS
3431 014052 017           .BYTE 17     ;SECTOR 17
3432 014053 000           .BYTE 0      ;TRACK 0
3433 014054 000312          312         ;CYLINDER 312
3434 014056 004437 037526    JSR R4,GENCOM ;GENERATE DATA
3435 014062 000004          4           ;PATTERN 4
3436 014064 000400          400         ;400 WORDS
3437
3438 014066 104417          TLOADRK      ;LOAD RK REGS
3439 014070 104424          TWAT32      ;WAIT FOR INTERRUPT
3440 014072 104002          ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
3441

```

```

3442 014074 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
3443 014076 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3444
3445 014100 004437 033324    JSR R4,LRLOAD    ;LOAD "L" REGS
3446 014104 000121          RDDATA          ;RDDATA
3447 014106 177675          -103           ;-103 WORDS
3448 014110 056446          IBUFF          ;IBUFF IS BUFF ADDRESS
3449 014112 017             .BYTE 17        ;SECTOR 17
3450 014113 000             .BYTE 0         ;TRACK 0
3451 014114 000312          312           ;CYLINDER 312
3452 014116 004437 037526    JSR R4,GENCOM
3453 014122 002007          2007          ;CLEAR IBUFF
3454 014124 000400          400
3455
3456 014126 104417          TLOADRK        ;LOAD RK REGS
3457 014130 104424          TWAT32        ;WAIT FOR INTERRUPT
3458 014132 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
3459 014134 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
3460 014136 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3461
3462 014140 012700 060654    MOV #OBUFF+206,RO ;AFTER THE LAST 205 WORDS OF
3463 014144 012701 000275    MOV #275,R1     ;THE OUTPUT BUFFER TO ALL ONES.
3464 014150 012720 177777    15: MOV #-1,(RO)+ ;THESE SHOULD ALL BE ONES IN
3465 014154 005301          DEC R1         ;IBUFF BECAUSE THE PARTIAL
3466 014156 001374          BNE 15        ;READ FILLED ONLY 103 WORDS.
3467 014160 004437 037526    JSR R4,GENCOM  ;GO COMPARE IBUFF & OBUFF
3468 014164 100000          100000
3469 014166 000400          400           ;ALL 400 WORDS
3470 014170 000413          BR 35         ;NO ERRORS-EXIT
3471 014172 104015          ERROR 15      ;REPORT FIRST COMPARE ERROR
3472
3473 014174 013700 001634    MOV ERLMT,RO   ;GET ERROR LIMIT
3474 014200 005300 64$: DEC RO        ;DECREMENT COUNT
3475 014202 001406          BEQ 65$       ;IF ZERO - EXIT
3476 014204 004437 037526    JSR R4,GENCOM ;CONTINUE DATA COMPARE
3477 014210 040000          40000
3478 014212 000402          3R 65$       ;NO MORE ERRORS - EXIT
3479 014214 104016          ERROR 16      ;REPORT NEXT ERROR
3480 014216 000770          BR 64$        ;LOOP
3481 014220 65$:
3482
3483 014220 3$:
3484
3485 *****
3486 *TEST 41 WRITE DATA WITH NON-EXISTENT MEMORY
3487 *
3488 * ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
3489 * MAKE SURE NON-EXISTENT MEMORY SETS.
3490 *****
3491 014220 000004          TST41: SCOPE
3492 014222 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
3493 014230 104416          TSSINIT       ;CLEAR SUBSYSTEM
3494 014232 104003          ERROR 3       ;BAD INIT ERROR
3495
3496 014234 004437 033324    JSR R4,LRLOAD ;LOAD "L" REG
3497 014240 001523          BA16!BA17!WRDATA ;BA16 & 17 SET WITH WRITE DATA
    
```

3498 014242 177777  
3499 014244 176000  
3500 014246 013  
3501 014247 000  
3502 014250 000312  
3503  
3504 014252 104417  
3505 014254 104423  
3506 014256 104002  
3507 014260 104422  
3508 014262 000000  
3509 014264 000000  
3510 014266 000040  
3511 014270 104004

-1 ;WORD COUNT OF 1  
176000 ;BUFF ADDRESS=IO PAGE BASE  
.BYTE 13 ;SECT 13  
.BYTE 0 ;TRACK 0  
312 ;CYLINDER 312  
TLOADRK ;LOAD RK REGS  
TWAT16 ;WAIT FOR INTERRUPT  
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR  
TCHKWE ;CHECK OPERATION WITH ERROR  
0  
0  
NEMERR ;NON-EXISTENT MEMORY ERROR  
ERROR 4 ;OR5,6,7 ;REPORT ANY DISCREPANCIES

\*\*\*\*\*  
\*TEST 42 READ DATA WITH NON-EXISTENT MEMORY  
\*\*\*\*\*

ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000.  
MAKE SURE NON-EXISTENT MEMORY SETS.

\*\*\*\*\*

3519 014272 000004  
3520 014274 012737 000062 001262  
3521 014302 104416  
3522 014304 104003

TST42: SCOPE  
MOV #50.,\$TIMES ;DO 50. ITERATIONS  
TSSINIT ;CLEAR SUBSYSTEM  
ERROR 3 ;BAD INIT ERROR

3524 014306 004437 033324  
3525 014312 001521  
3526 014314 177777  
3527 014316 176000  
3528 014320 013  
3529 014321 000  
3530 014322 000312

JSR R4,LRLOAD ;LOAD "L" REG  
BA16!BA17!RDATA ;BA16 & 17 WITH READ DATA  
-1 ;WORD COUNT OF 1  
176000 ;BUFF ADDRESS=IO PAGE BASE  
.BYTE 13 ;SECTOR 13  
.BYTE 0 ;TRACK 0  
312 ;CYL 312

3531  
3532 014324 104417  
3533 014326 104424  
3534 014330 104002  
3535 014332 104422  
3536 014334 000000  
3537 014336 000000  
3538 014340 000040  
3539 014342 104004

TLOADRK ;LOAD RK REGS  
TWAT32 ;WAIT FOR INTERRUPT  
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR  
TCHKWE ;CHECK OPERATION WITH ERRORS  
0  
0  
NEMERR ;NON-EXISTENT MEMORY ERROR  
ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES

\*\*\*\*\*  
\*TEST 43 UNIBUS PARITY ERROR  
\*\*\*\*\*

INITIALIZE A MEMORY LOCATION WITH BAD  
PARITY. ISSUE A WRITE DATA OF 400 WORDS  
STARTING AT A LOCATION 110 WORDS BEFORE  
THE LOCATION WITH BAD PARITY. MAKE SURE  
THAT UNIBUS PARITY ERROR SETS.

NOTE: THIS TEST IS ONLY EXECUTED IF  
MEMORY PARITY OPTION EXISTS FOR  
BUFFER.

\*\*\*\*\*

3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553

```

3554 014344 000004          TST43: SCOPE
3555 014346 012737 000062 001262  MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3556 014354 104416          TSSINIT  ;;CLEAR SUBSYSTEM
3557 014356 104003          ERROR    3                ;;BAD INIT ERROR
3558
3559 014360 032737 000200 001656  BIT      #PARBK0,OPTFLG  ;;TEST IF PARITY OPTION PRESENT
3560 014366 001012          BNE     1$                ;;YES-SKIP
3561 014370 032737 000004 001656  BIT      #MEMPYB,OPTFLG  ;;TEST IF PARITY OPTION REPORTED
3562 014376 001123          BNE     2$                ;;NO-SKIP TO EXIT
3563 014400 104401 046312          TYPE    ,OPRD10          ;;PRINT BYPASS MESSAGE
3564 014404 052737 000004 001656  BIS     #MEMPYB,OPTFLG  ;;SET OPTION REPORTED BIT
3565 014412 000515          BR      2$                ;;SKIP TO EXIT
3566
3567 014414          1$:
3568 014414 004437 033324          JSR     R4,LRLOAD        ;;LOAD "L" REGS
3569 014420 000123          WRDATA  ;;WRDATA
3570 014422 177400          -400    ;;-400 WORDS
3571 014424 060446          OBUFF   ;;OBUFF IS BUFF ADDRESS
3572 014426 010          .BYTE  10                ;;SECTOR 10
3573 014427 000          .BYTE  0                  ;;TRACK 0
3574 014430 000312          312     ;;CYLINDER 312
3575
3576 014432 005077 165242          CLR     @MMCSR1 ;CLEAR PARITY IE
3577 014436 032737 000100 001656  BIT     #PARBK1,OPTFLG  ;;IS BANK 1 AVAIL
3578 014444 001402          BEQ    3$                ;;NO - SKIP
3579 014446 005077 165230          CLR     @MMCSR2 ;CLEAR IE FOR BANK1
3580
3581 014452 004437 037526          3$: JSR     R4,GENCOM        ;;GENERATE DATA
3582 014456 000005          5       ;;PATTERN 5
3583 014460 000400          400    ;;400 WORDS
3584
3585 014462 012746 000340          MOV     #PR7,-(SP)       ;;PUT PRIORITY 7 ON STACK
3586 014466 012746 014474          MOV     #10$,-(SP)      ;;PUT ADDRESS ON STACK
3587 014472 000002          RTI    ;;SET PRI
3588 014474          10$:
3589
3590 014474 032737 000100 001656  BIT     #PARBK1,OPTFLG  ;;TEST IF PARITY BANK 1 AVAIL
3591 014502 001403          BEQ    4$                ;;NO - SKIP
3592 014504 012777 000004 165170          MOV     #BIT2,@MMCSR2   ;;SET WRITE WRONG PARITY ENABLE
3593 014512 012777 000004 165160          MOV     #BIT2,@MMCSR1   ;;SET WRITE WRONG PARITY BIT
3594 014520 012737 060674 060672          MOV     #OBUFF+226,OBUFF+224
3595 014526 012737 060672 060670          MOV     #OBUFF+224,OBUFF+222 ;;WRITE ALL ONES INTO BUFFER W/BAD PARITY
3596 014534 012777 000001 165136          MOV     #BIT0,@MMCSR1   ;;CLEAR CONTROL BIT, SET IE BIT
3597 014542 032737 000100 001656  BIT     #PARBK1,OPTFLG  ;;TEST IF BANK 1 AVAIL
3598 014550 001403          BEQ    5$                ;;NO SKIP
3599 014552 012777 000001 165122          MOV     #BIT0,@MMCSR2   ;;CLEAR CONTROL BIT, SET IE
3600
3601 014560 013746 001622          5$: MOV     RKPRI,-(SP)      ;;SET OLD PRIORITY
3602 014564 012746 014572          MOV     #11$,-(SP)      ;;SET ADDRESS
3603 014570 000002          RTI    ;;RESTORE PRI
3604 014572          11$:
3605
3606 014572 104417          TLOADRK ;;LOAD RK REGS
3607 014574 104424          TWAT32  ;;WAIT FOR INTERRUPT
3608 014576 104002          ERROR  2                ;;TO SLOW/NOT COMPLETE ERROR
3609 014600 005077 165074          CLR     @MMCSR1 ;TURN OFF CS1

```



```

3610 014604 032737 000100 001656 BIT #PARBK1,OPTFLG ;TEST IF BANK 1 PARITY PRESENT
3611 014612 001402 BEQ 6$ ;NO SKIP
3612 014614 005077 165062 CLR @MMCSR2 ;TURN OFF BANK 1
3613 014620 005037 060672 6$: CLR OBUFF+224
3614 014624 005037 060670 CLR OBUFF+222 ;CLEAR BAD PARITY ERROR
3615
3616 014630 004737 032434 JSR PC,OPTTST ;RESET OPTIONS
3617
3618 014634 104422 TCHKWE ;CHECK OPERATION WITH ERROR
3619 014636 000000 0
3620 014640 000000 0
3621 014642 000100 UPERR ;UNIBUS PARITY ERROR
3622 014644 104004 ERROR 4; OR 5,6,7 ;REPORT ALL DISCREPANCIES
3623
3624 014646 2$:
3625
3626 .SBTTL **MULTIPLE SECTOR OPERATIONS
3627
3628 ;*****
3629 ;*TEST 44 TWO SECTOR WRITE DATA (PART 1)
3630 ;*
3631 ;* ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
3632 ;* TRACK 0, SECTOR 0. READ DATA BACK ONE SECTOR
3633 ;* AT A TIME AND MAKE SURE IT IS CORRECT.
3634 ;*
3635 ;*****
3636 014646 000004 TST44: SCOPE
3637 014650 012737 000062 001262 MOV #50.,$TIMES ;:DO 50. ITERATIONS
3638 014656 104416 TSSINIT ;:CLEAR SUBSYSTEM
3639 014660 104003 ERROR 3 ;:BAD INIT ERROR
3640
3641 014662 004437 033324 JSR R4,LRLOAD ;:LOAD "L" REGS
3642 014666 000123 WRDATA ;:WRDATA
3643 014670 177000 -1000 ;:-1000 WORDS
3644 014672 060446 OBUFF ;:OBUFF IS BUFF ADDRESS
3645 014674 000 .BYTE 0 ;:SECTOR 0
3646 014675 000 .BYTE 0 ;:TRACK 0
3647 014676 000312 312 ;:CYLINDER 312
3648
3649 014700 004437 037526 JSR R4,GENCOM ;:GENERATE DATA
3650 014704 000015 15 ;:PATTERN 15
3651 014706 001000 1000 ;:1000 WORDS
3652
3653 014710 104417 TLOADRK ;:LOAD RK REGS
3654 014712 104424 TWAT32 ;:WAIT FOR INTERRUPT
3655 014714 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
3656
3657 014716 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
3658 014720 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3659
3660 014722 004437 037526 JSR R4,GENCOM ;:CLEAR Ibuff
3661 014726 002007 2007 ;:TO ALL 1'S
3662 014730 001000 1000
3663
3664 014732 004437 033324 JSR R4,LRLOAD ;:LOAD "L" REGS
3665 014736 000121 RDDATA ;:RDDATA
    
```

```

3666 014740 177400          -400          ; -400 WORDS
3667 014742 056446          Ibuff         ; Ibuff IS buff ADDRESS
3668 014744          000          .BYTE 0       ; SECTOR 0
3669 014745          000          .BYTE 0       ; TRACK 0
3670 014746 000312          312          ; CYLINDER 312
3671
3672 014750 104417          TLOADRK       ; LOAD RK REGS
3673 014752 104424          TWAT32        ; WAIT FOR INTERRUPT
3674 014754 104002          ERROR 2      ; TO SLOW/NOT COMPLETE ERROR
3675
3676 014756 104421          TCHKOP        ; CHECK OPERATION FOR ANY ERRORS
3677 014760 104004          ERROR 4 ;OR 5, 6, 7, 10 ; REPORT ALL ERRORS
3678
3679 014762 004437 033324    JSR R4,LRLOAD ; LOAD "L" REGS
3680 014766 000121          RDDATA        ; RDDATA
3681 014770 177400          -400          ; -400 WORDS
3682 014772 057446          Ibuff+1000    ; Ibuff+1000 IS buff ADDRESS
3683 014774          001          .BYTE 1       ; SECTOR 1
3684 014775          000          .BYTE 0       ; TRACK 0
3685 014776 000312          312          ; CYLINDER 312
3686
3687 015000 104417          TLOADRK       ; LOAD RK REGS
3688 015002 104424          TWAT32        ; WAIT FOR INTERRUPT
3689 015004 104002          ERROR 2      ; TO SLOW/NOT COMPLETE ERROR
3690
3691 015006 104421          TCHKOP        ; CHECK OPERATION FOR ANY ERRORS
3692 015010 104004          ERROR 4 ;OR 5, 6, 7, 10 ; REPORT ALL ERRORS
3693
3694 015012 004437 037526    JSR R4,GENCOM ; COMPARE DATA
3695 015016 100000          100000        ;
3696 015020 001000          1000          ; 1000 WORDS
3697 015022 000413          BR 2$         ; NO MISCOMPARES-EXIT
3698 015024 104015          ERROR 15      ; REPORT FIRST ERROR
3699
3700 015026 013700 001634    MOV ERLMT,RO  ; GET ERROR LIMIT
3701 015032 005300          64$: DEC RO   ; DECREMENT COUNT
3702 015034 001406          BEQ 65$      ; IF ZERO - EXIT
3703 015036 004437 037526    JSR R4,GENCOM ; CONTINUE DATA COMPARE
3704 015042 040000          40000        ;
3705 015044 000402          BR 65$      ; NO MORE ERRORS - EXIT
3706 015046 104016          ERROR 16     ; REPORT NEXT ERROR
3707 015050 000770          BR 64$      ; LOOP
3708 015052          65$:
3709
3710 015052          2$:
3711          ; *****
3712          ; *TEST 45 TWO SECTOR WRITE DATA (PART 2)
3713          ; *
3714          ; * ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
3715          ; * TRACK 0, SECTOR 23. READ DATA BACK ONE SECTOR
3716          ; * AT A TIME AND MAKE SURE A MID-TRANSFER
3717          ; * SEEK DID NOT TAKE PLACE.
3718          ; *
3719          ; *****
3720 015052 000004          TST45: SCOPE
3721 015054 012737 000062 001262 MOV #50.,$TIMES ; DO 50. ITERATIONS

```

```

3722 015062 104416          TSSINIT          ;CLEAR SUBSYSTEM
3723 015064 104003          ERROR 3          ;BAD INIT ERROR
3724
3725 015066 004437 033324   JSR R4,LRLOAD    ;LOAD "L" REGS
3726 015072 000123          WRDATA          ;WRDATA
3727 015074 177000          -1000          ;-1000 WORDS
3728 015076 060446          OBUFF          ;OBUFF IS BUFF ADDRESS
3729 015100 023             .BYTE 23        ;SECTOR 23
3730 015101 000             .BYTE 0         ;TRACK 0
3731 015102 000312          312           ;CYLINDER 312
3732
3733 015104 004437 037526   JSR R4,GENCOM    ;GENERATE DATA
3734 015110 000016          16            ;PATTERN 16
3735 015112 001000          1000          ;1000 WORDS
3736
3737 015114 104417          TLOADRK        ;LOAD RK REGS
3738 015116 104424          TWAT32        ;WAIT FOR INTERRUPT
3739 015120 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
3740
3741 015122 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
3742 015124 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3743
3744          ;
3745          ;
3746          ;
3747          ;
3748 015126 004437 033324   JSR R4,LRLOAD    ;LOAD "L" REGS
3749 015132 000121          RDDATA        ;RDDATA
3750 015134 177400          -400          ;-400 WORDS
3751 015136 056446          IBUFF        ;IBUFF IS BUFF ADDRESS
3752 015140 023             .BYTE 23        ;SECTOR 23
3753 015141 000             .BYTE 0         ;TRACK 0
3754 015142 000312          312           ;CYLINDER 312
3755
3756 015144 004437 037526   JSR R4,GENCOM    ;CLEAR IBUFF TO ALL ONES
3757 015150 002007          2007
3758 015152 001000          1000
3759
3760 015154 104417          TLOADRK        ;LOAD RK REGS
3761 015156 104424          TWAT32        ;WAIT FOR INTERRUPT
3762 015160 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
3763
3764 015162 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
3765 015164 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3766
3767 015166 004437 033324   JSR R4,LRLOAD    ;LOAD "L" REGS
3768 015172 000121          RDDATA        ;RDDATA
3769 015174 177400          -400          ;-400 WORDS
3770 015176 057446          IBUFF+1000    ;IBUFF+1000 IS BUFF ADDRESS
3771 015200 024             .BYTE 24        ;SECTOR 24
3772 015201 000             .BYTE 0         ;TRACK 0
3773 015202 000312          312           ;CYLINDER 312
3774
3775 015204 104417          TLOADRK        ;LOAD RK REGS
3776 015206 104424          TWAT32        ;WAIT FOR INTERRUPT
3777 015210 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR

```

```

3778
3779 015212 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
3780 015214 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3781
3782 015216 004437 037526   JSR R4,GENCOM   ;COMPARE DATA
3783 015222 100000          100000
3784 015224 001000          1000           ;1000 WORDS
3785 015226 000413          BR 1$          ;NO ERRORS-SKIP
3786 015230 104015          ERROR 15       ;REPORT FIRST ERROR
3787
3788 015232 013700 001634   MOV ERR,MT,RO  ;GET ERROR LIMIT
3789 015236 005300          DEC RO         ;DECREMENT COUNT
3790 015240 001406          SEQ 65$       ;IF ZERO - EXIT
3791 015242 004437 037526   JSR R4,GENCOM ;CONTINUE DATA COMPARE
3792 015246 040000          40000
3793 015250 000402          BR 65$       ;NO MORE ERRORS - EXIT
3794 015252 104016          ERROR 16      ;REPORT NEXT ERROR
3795 015254 000770          BR 64$       ;LOOP
3796 015256
3797 015256
3798
3799
3800
3801
3802
3803
3804
3805
3806 015256 000004          ;*****
3807 015260 012737 000062 001262 ;TEST 46      TWO SECTOR WRITE DATA (PART 3)
3808 015266 104416          ;*
3809 015270 104003          ;*
3810
3811 015272 004437 033324   ;*
3812 015276 000123          ;*
3813 015300 177377          ;*
3814 015302 060446          ;*
3815 015304 010           ;*
3816 015305 000           ;*
3817 015306 000312          ;*
3818
3819 015310 004437 037526   ;*
3820 015314 000002          ;*
3821 015316 001000          ;*
3822
3823 015320 104417          ;*
3824 015322 104424          ;*
3825 015324 104002          ;*
3826
3827 015326 104421          ;*
3828 015330 104004          ;*
3829
3830 015332 012700 061450   ;*
3831 015336 012701 000377   ;*
3832 015342 005020          ;*
3833 015344 005301          ;*

```

64\$:

65\$:

1\$:

```

;*****
;TEST 46      TWO SECTOR WRITE DATA (PART 3)
;*
;*
;*
;*
;*
;*
;*****

```

1\$46:

```

SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-401 ;-401 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 10 ;SECTOR 10
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312
JSR R4,GENCOM ;GENERATE DATA
2 ;PATTERN 2
1000 ;1000 WORDS
TLOADRK ;LOAD RK REGS
TWAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
CLEAR LAST 377 WORDS OF OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
MOV #OBUFF+1002,RO ;GET STARTING ADDRESS TO BE CLEARED
MOV #377 R1 ;NUMBER OF WORDS
CLR (RO) ;CLEAR WORD
DEC R1 ;DEC COUNT

```

```

3834 015346 001375      BNE      1$      ;LOOP UNTIL COUNT ZERO
3835 015350 004437 037526 JSR      R4,GENCOM ;CLEAR IBUFF TO ONES
3836 015354 002007      2007
3837 015356 001000      1000
3838
3839 015360 004437 033324 JSR      R4,LRLOAD ;LOAD "L" REGS
3840 015364 000121      RDDATA ;RDDATA
3841 015366 177400      -400 ; -400 WORDS
3842 015370 056446      IBUFF ;IBUFF IS BUFF ADDRESS
3843 015372 010 ;.BYTE 10 ;SECTOR 10
3844 015373 000 ;.BYTE 0 ;TRACK 0
3845 015374 000312      312 ;CYLINDER 312
3846
3847 015376 104417      TLOADRK ;LOAD RK REGS
3848 015400 104424      TWAT32 ;WAIT FOR INTERRUPT
3849 015402 104002      ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3850
3851 015404 104421      TCHKOP ;CHECK OPERATION FOR ANY ERRORS
3852 015406 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3853 015410 004437 033324 JSR      R4,LRLOAD ;LOAD "L" REGS
3854 015414 000121      RDDATA ;RDDATA
3855 015416 177400      -400 ; -400 WORDS
3856 015420 057446      IBUFF+1000 ;IBUFF+1000 IS BUFF ADDRESS
3857 015422 011 ;.BYTE 11 ;SECTOR 11
3858 015423 000 ;.BYTE 0 ;TRACK 0
3859 015424 000312      312 ;CYLINDER 312
3860
3861 015426 104417      TLOADRK ;LOAD RK REGS
3862 015430 104424      TWAT32 ;WAIT FOR INTERRUPT
3863 015432 104002      ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3864
3865 015434 004437 037526 JSR      R4,GENCOM ;DATA COMPARE
3866 015440 100000      100000
3867 015442 001000      1000 ;1000 WORDS
3868 015444 000413      BR      2$ ;NO ERROR-SKIP
3869 015446 104015      ERROR 15 ;REPORT FIRST ERROR
3870
3871 015450 013700 001634 MOV      ERRLMT,R0 ;GET ERROR LIMIT
3872 015454 005300      64$: DEC      R0 ;DECREMENT COUNT
3873 015456 001406      BEQ      65$ ;IF ZERO - EXIT
3874 015460 004437 037526 JSR      R4,GENCOM ;CONTINUE DATA COMPARE
3875 015464 040000      40000
3876 015466 000402      BR      65$ ;NO MORE ERRORS - EXIT
3877 015470 104016      ERROR 16 ;REPORT NEXT ERROR
3878 015472 000770      BR      64$ ;LOOP
3879 015474
3880
3881 015474      2$:
3882 *****
3883 ;*TEST 47 MID-TRANSFER SEEK ON WRITE (PART 1)
3884 ;*
3885 ;*
3886 ;* ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
3887 ;* TRACK 0, SECTOR 25. READ DATA BACK ONE SECTOR
3888 ;* AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
3889 ;* DID TAKE PLACE.

```

```

3890
3891 015474 000004
3892 015476 012737 000062 001262
3893 015504 104416
3894 015506 104003
3895
3896 015510 004437 033324
3897 015514 000123
3898 015516 177000
3899 015520 060446
3900 015522 025
3901 015523 000
3902 015524 000312
3903
3904 015526 004437 037526
3905 015532 000003
3906 015534 001000
3907
3908 015536 104417
3909 015540 104425
3910 015542 104002
3911
3912 015544 104421
3913 015546 104004
3914
3915
3916 015550 004437 037526
3917 015554 002007
3918 015556 001000
3919
3920 015560 004437 033324
3921 015564 000121
3922 015566 177400
3923 015570 056446
3924 015572 025
3925 015573 000
3926 015574 000312
3927
3928 015576 104417
3929 015600 104425
3930 015602 104002
3931
3932 015604 104421
3933 015606 104004
3934
3935 015610 004437 033324
3936 015614 000121
3937 015616 177400
3938 015620 057446
3939 015622 000
3940 015623 001
3941 015624 000312
3942
3943 015626 104417
3944 015630 104425
3945 015632 104002

```

:\*\*\*\*\*  
†ST47: SCOPE  
MOV #50.,\$TIMES ;:DO 50. ITERATIONS  
TSSINIT ;:CLEAR SUBSYSTEM  
ERROR 3 ;:BAD INIT ERROR  
  
JSR R4,LRLOAD ;:LOAD "L" REGS  
WRDATA ;:WRDATA  
-1000 ;:-1000 WORDS  
OBUFF ;:OBUFF IS BUFF ADDRESS  
.BYTE 25 ;:SECTOR 25  
.BYTE 0 ;:TRACK 0  
312 ;:CYLINDER 312  
  
JSR R4,GENCOM ;:GENERATE DATA  
3 ;:PATTERN 3  
1000 ;:1000 WORDS  
  
TLOADRK ;:LOAD RK REGS  
TWAT48 ;:WAIT FOR INTERRUPT  
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR  
  
TCHKOP ;:CHECK OPERATION FOR ANY ERRORS  
ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS  
A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A  
MID-TRANSFER HEAD SWITCH DID NOT OCCUR.  
  
JSR R4,GENCOM  
2007  
1000  
  
JSR R4,LRLOAD ;:LOAD "L" REGS  
RDATA ;:RDATA  
-400 ;:-400 WORDS  
IBUFF ;:IBUFF IS BUFF ADDRESS  
.BYTE 25 ;:SECTOR 25  
.BYTE 0 ;:TRACK 0  
312 ;:CYLINDER 312  
  
TLOADRK ;:LOAD RK REGS  
TWAT48 ;:WAIT FOR INTERRUPT  
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR  
  
TCHKOP ;:CHECK OPERATION FOR ANY ERRORS  
ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS  
  
JSR R4,LRLOAD ;:LOAD "L" REGS  
RDATA ;:RDATA  
-400 ;:-400 WORDS  
IBUFF+1000 ;:IBUFF+1000 IS BUFF ADDRESS  
.BYTE 0 ;:SECTOR 0  
.BYTE 1 ;:TRACK 1  
312 ;:CYLINDER 312  
  
TLOADRK ;:LOAD RK REGS  
TWAT48 ;:WAIT FOR INTERRUPT  
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR

```

3946
3947 015634 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
3948 015636 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
3949
3950 015640 004437 037526    JSR R4,GENCOM   ;COMPARE DATA
3951 015644 100000          100000
3952 015646 001000          1000           ;1000 WORDS
3953 015650 000413          BR 1$          ;NO ERRORS-SKIP
3954 015652 104015          ERROR 15       ;REPORT FIRST ERROR
3955
3956 015654 013700 001634    MOV ERLMT,RO   ;GET ERROR LIMIT
3957 015660 005300          64$: DEC RO     ;DECREMENT COUNT
3958 015662 001406          BEQ 65$       ;IF ZERO - EXIT
3959 015664 004437 037526    JSR R4,GENCOM ;CONTINUE DATA COMPARE
3960 015670 040000          40000
3961 015672 000402          BR 65$       ;NO MORE ERRORS - EXIT
3962 015674 104016          ERROR 16      ;REPORT NEXT ERROR
3963 015676 000770          BR 64$       ;LOOP
3964 015700
3965
3966 015700          1$:
3967          ;*****
3968          ;TEST 50 MID-TRANSFER SEEK ON WRITE (PART 2)
3969          ;*
3970          ;* ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
3971          ;* TRACK 2, SECTOR 25. READ DATA BACK ONE SECTOR
3972          ;* AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
3973          ;* DID TAKE PLACE.
3974          ;*
3975          ;*****
3976 015700 000004          1$T50: SCOPE
3977 015702 012737 000062 001262 MOV #50.,$TIMES ;:DO 50. ITERATIONS
3978 015710 104416          TSSINIT       ;:CLEAR SUBSYSTEM
3979 015712 104003          ERROR 3       ;:BAD INIT ERROR
3980
3981 015714 004437 033324    JSR R4,LRLOAD  ;:LOAD "L" REGS
3982 015720 000123          WRDATA       ;:WRDATA
3983 015722 177000          -1000       ;:-1000 WORDS
3984 015724 060446          OBUFF        ;:OBUFF IS BUFF ADDRESS
3985 015726 025           .BYTE 25     ;:SECTOR 25
3986 015727 002           .BYTE 2       ;:TRACK 2
3987 015730 000312          312         ;:CYLINDER 312
3988
3989 015732 004437 037526    JSR R4,GENCOM ;:GENERATE DATA
3990 015736 000004          4           ;:PATTERN 4
3991 015740 001000          1000        ;:1000 WORDS
3992
3993 015742 104417          TLOADRK      ;:LOAD RK REGS
3994 015744 104426          TWAT64      ;:WAIT FOR INTERRUPT
3995 015746 104002          ERROR 2     ;:TO SLOW/NOT COMPLETE ERROR
3996          ; A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A
3997          ; MID-TRANSFER SEEK DID NOT OCCUR.
3998 015750 104421          TCHKOP      ;:CHECK OPERATION FOR ANY ERRORS
3999 015752 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4000
4001 015754 004437 037526    JSR R4,GENCOM ;:CLEAR Ibuff TO ALL ONES
    
```

```

4002 015760 002007          2007
4003 015762 001000          1000
4004
4005 015764 004437 033324   JSR      R4,LRLOAD      ;LOAD "L" REGS
4006 015770 000121          RDDATA      ;RDDATA
4007 015772 177400          -400        ;-400 WORDS
4008 015774 056446          Ibuff      ;IBUFF IS BUFF ADDRESS
4009 015776          025      .BYTE      25          ;SECTOR 25
4010 015777          002      .BYTE      2          ;TRACK 2
4011 016000 000312          312        ;CYLINDER 312
4012
4013 016002 104417          TLOADRK     ;LOAD RK REGS
4014 016004 104425          TWAT48      ;WAIT FOR INTERRUPT
4015 016006 104002          ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
4016
4017 016010 104421          TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
4018 016012 104004          ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4019
4020 016014 004437 033324   JSR      R4,LRLOAD      ;LOAD "L" REGS
4021 016020 000121          RDDATA      ;RDDATA
4022 016022 177400          -400        ;-400 WORDS
4023 016024 057446          Ibuff+1000 ;IBUFF+1000 IS BUFF ADDRESS
4024 016026          000      .BYTE      0          ;SECTOR 0
4025 016027          000      .BYTE      0          ;TRACK 0
4026 016030 000313          313        ;CYLINDER 313
4027
4028 016032 104417          TLOADRK     ;LOAD RK REGS
4029 016034 104425          TWAT48      ;WAIT FOR INTERRUPT
4030 016036 104002          ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
4031
4032 016040 104421          TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
4033 016042 104004          ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4034
4035 016044 004437 037526   JSR      R4,GENCOM     ;COMPARE DATA
4036 016050 100000          100000     ;1000 WORDS
4037 016052 001000          1000        ;NO MISCOMPARES-SKIP
4038 016054 000413          BR         15          ;REPORT 1ST ERROR
4039 016056 104015          ERROR      15
4040
4041 016060 013700 001634   MOV      ERRMT,RO      ;GET ERROR LIMIT
4042 016064 005300          64$: DEC      RO        ;DECREMENT COUNT
4043 016066 001406          BEQ      65$          ;IF ZERO - EXIT
4044 016070 004437 037526   JSR      R4,GENCOM     ;CONTINUE DATA COMPARE
4045 016074 040000          40000      ;NO MORE ERRORS - EXIT
4046 016076 000402          BR         65$          ;REPORT NEXT ERROR
4047 016100 104016          ERROR      16
4048 016102 000770          BR         64$          ;LOOP
4049 016104
4050
4051 016104          1$:
4052          ;*****
4053          ;*TEST 51      TWO SECTOR READ DATA (PART 1)
4054          ;*
4055          ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
4056          ;*      TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS
4057          ;*      READ.

```



```

4058
4059
4060
4061
4062
4063 016104 000004
4064 016106 012737 000062 001262
4065 016114 104416
4066 016116 104003
4067
4068
4069
4070
4071 016120 004437 037526
4072 016124 000015
4073 016126 001000
4074
4075 016130 004437 037526
4076 016134 002007
4077 016136 001000
4078
4079 016140 004437 033324
4080 016144 000121
4081 016146 177000
4082 016150 056446
4083 016152 000
4084 016153 000
4085 016154 000312
4086
4087 016156 104417
4088 016160 104424
4089 016162 104002
4090
4091 016164 104421
4092 016166 104004
4093
4094 016170 004437 037526
4095 016174 100000
4096 016176 001000
4097 016200 000413
4098 016202 104015
4099
4100 016204 013700 001634
4101 016210 005300
4102 016212 001406
4103 016214 004437 037526
4104 016220 040000
4105 016222 000402
4106 016224 104016
4107 016226 000770
4108 016230
4109
4110 016230
4111
4112
4113

```

```

;*
;* NOTE: TWO SECTOR WRITE DATA (PART 1) MUST BE
;* EXECUTED BEFORE THIS TEST.
;*
*****
TST51: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA (PART 1)
; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA PART 1
JSR R4,GENCOM ;GENERATE DATA
15 ;PATTERN 15
1000 ;1000 WORDS

JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
2007
1000

JSR R4,LRLOAD ;LOAD "L" REGS
RDDATA ;RDDATA
-1000 ;-1000 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

TLOADRK ;LOAD RK REGS
TWAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

JCS R4,GENCOM ;COMPARE DATA
100000
1000 ;1000 WORDS
BR 15 ;NO MISCOMPARES-SKIP
ERROR 15

MOV ERRLMT,R0 ;GET ERROR LIMIT
64$: DEC R0 ;DECREMENT COUNT
BEQ 65$ ;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 65$ ;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 64$ ;LOOP

65$:
1$:
*****
;TEST 52 TWO SECTOR READ DATA (PART 2)
;*
```

```

4114      ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
4115      ;*      TRACK 0, SECTOR 23. VERIFY THAT CORRECT DATA IS
4116      ;*      READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.
4117      ;*
4118      ;*      NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE
4119      ;*      EXECUTED BEFORE THIS TEST.
4120      ;*
4121      ;*****
4122 016230 000004      †ST52: SCOPE
4123 016232 012737 000062 001262      MOV #50.,$TIMES      ;;DO 50. ITERATIONS
4124 016240 104416      TSSINIT      ;CLEAR SUBSYSTEM
4125 016242 104003      ERROR 3      ;BAD INIT ERROR
4126
4127      ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)
4128
4129 016244 004437 037526      JSR R4,GENCOM      ;GENERATE DATA
4130 016250 000016      16      ;PATTERN 16
4131 016252 001000      1000      ;1000 WORDS
4132
4133 016254 004437 037526      JSR R4,GENCOM      ;CLEAR Ibuff TO ALL ONES
4134 016260 002007      2007
4135 016262 001000      1000
4136 016264 004437 033324      JSR R4,LRLoad      ;LOAD "L" REGS
4137 016270 000121      RDATA      ;RDATA
4138 016272 177000      -1000      ;-1000 WORDS
4139 016274 056446      Ibuff      ;IBuff IS Buff ADDRESS
4140 016276 023      .BYTE 23      ;SECTOR 23
4141 016277 000      .BYTE 0      ;TRACK 0
4142 016300 000312      312      ;CYLINDER 312
4143
4144 016302 104417      TLOADRK      ;LOAD RK REGS
4145 016304 104424      TWAT32      ;WAIT FOR INTERRUPT
4146 016306 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
4147
4148 016310 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
4149 016312 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4150
4151 016314 004437 037526      JSR R4,GENCOM      ;COMPARE DATA
4152 016320 100000      100000
4153 016322 001000      1000      ;1000 WORDS
4154 016324 000413      BR 15      ;NO MISCOMPARES-SKIP
4155 016326 104015      ERROR 15      ;REPORT 1ST ERROR
4156
4157 016330 013700 001634      MOV ERRlMT,R0      ;GET ERROR LIMIT
4158 016334 005300      64$: DEC R0      ;DECREMENT COUNT
4159 016336 001406      BEQ 65$      ;IF ZERO - EXIT
4160 016340 004437 037526      JSR R4,GENCOM      ;CONTINUE DATA COMPARE
4161 016344 040000      40000
4162 016346 000402      BR 65$      ;NO MORE ERRORS - EXIT
4163 016350 104016      ERROR 16      ;REPORT NEXT ERROR
4164 016352 000770      BR 64$      ;LOOP
4165 016354      65$:
4166
4167 016354      1$:
4168      ;*****
4169      ;*TEST 53 TWO SECTOR READ DATA (PART 3)

```

4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179  
4180  
4181  
4182  
4183  
4184  
4185  
4186  
4187  
4188  
4189  
4190  
4191  
4192  
4193  
4194  
4195  
4196  
4197  
4198  
4199  
4200  
4201  
4202  
4203  
4204  
4205  
4206  
4207  
4208  
4209  
4210  
4211  
4212  
4213  
4214  
4215  
4216  
4217  
4218  
4219  
4220  
4221  
4222  
4223  
4224  
4225

016354 000004  
016356 012737 000062 001262  
016364 104416  
016366 104003  
  
016370 004437 037526  
016374 000002  
016376 000401  
  
016400 004437 037526  
016404 002007  
016406 001000  
  
016410 004437 033324  
016414 000121  
016416 177377  
016420 056446  
016422 010  
016423 000  
016424 000312  
  
016426 104417  
016430 104424  
016432 104002  
  
016434 104421  
016436 104004  
  
016440 004437 037526  
016444 100000  
016446 000401  
016450 000413  
016452 104015  
  
016454 013700 001634  
016460 005300  
016462 001406  
016464 004437 037526  
016470 040000  
016472 000402  
016474 104016  
016476 000770  
016500  
016500

```

;*
;*   ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312,
;*   TRACK 0, SECTOR 10. VERIFY THAT ALL 401 WORDS
;*   ARE PLACED IN MEMORY.
;*
;*   NOTE: TWO SECTOR WRITE DATA (PART 3) MUST BE
;*   EXECUTED BEFORE THIS TEST.
;*
;*****
TST53: SCOPE
MOV #50.,$TIMES ;:DO 50. ITERATIONS
TSSINIT ;:CLEAR SUBSYSTEM
ERROR 3 ;:BAD INIT ERROR
;
; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)
;
JSR R4,GENCOM ;:GENERATE DATA
2 ;:PATTERN 2
401 ;:401 WORDS
;
JSR R4,GENCOM ;:CLEAR IBLFF TO ALL ONES
2007
1000
;
JSR R4,LRLOAD ;:LOAD "L" REGS
RDATA ;:RDATA
-401 ;:-401 WORDS
IBUFF ;:IBUFF IS BUFF ADDRESS
.BYTE 10 ;:SECTOR 10
.BYTE 0 ;:TRACK 0
312 ;:CYLINDER 312
;
TLOADRK ;:LOAD RK REGS
TWAT32 ;:WAIT FOR INTERRUPT
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
;
TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS
;
JSR R4,GENCOM ;:COMPARE DATA
100000
401 ;:401 WORDS
BR 15 ;:NO MISCOMPARES-SKIP
ERROR 15 ;:PRINT FIRST ERROR
;
MOV ERRLMT,R0 ;:GET ERROR LIMIT
64$: DEC R0 ;:DECREMENT COUNT
BEQ 65$ ;:IF ZERO - EXIT
JSR R4,GENCOM ;:CONTINUE DATA COMPARE
40000
BR 65$ ;:NO MORE ERRORS - EXIT
ERROR 16 ;:REPORT NEXT ERROR
BR 64$ ;:LOOP
;
65$:
1$:
;*****

```

4226  
4227  
4228  
4229  
4230  
4231  
4232  
4233  
4234  
4235  
4236  
4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259  
4260  
4261  
4262  
4263  
4264  
4265  
4266  
4267  
4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281

016500 000004  
016502 012737 000062 001262  
016510 104416  
016512 104003  
  
016514 004437 037526  
016520 000003  
016522 001000  
  
016524 004437 037526  
016530 002007  
016532 001000  
  
016534 004437 033324  
016540 000121  
016542 177000  
016544 056446  
016546 025  
016547 000  
016550 000312  
  
016552 104417  
016554 104424  
016556 104002  
  
016560 104421  
016562 104004  
  
016564 004437 037526  
016570 100000  
016572 001000  
016574 000413  
016576 104015  
  
016600 013700 001634  
016604 005300  
016606 001406  
016610 004437 037526  
016614 040000  
016616 000402  
016620 104016  
016622 000770  
016624  
016624

```

; *TEST 54 MID-TRANSFER SEEK ON READ (PART 1)
;
; ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
; TRACK 0, SECTOR 25. VERIFY THAT CORRECT DATA IS
; READ AND A MID-TRANSFER SEEK DOES OCCUR.
;
; NOTE: MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
; EXECUTED BEFORE THIS TEST.
;
; *****
; ST54: SCOPE
; MOV #50., $TIMES ; DO 50. ITERATIONS
; TSSINIT ; CLEAR SUBSYSTEM
; ERROR 3 ; BAD INIT ERROR
;
; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
; JSR R4, GENCOM ; GENERATE DATA
; 3 ; PATTERN 3
; 1000 ; 1000 WORDS
;
; JSR R4, GENCOM ; CLEAR Ibuff TO ALL ONES
; 2007
; 1000
;
; JSR R4, LRLOAD ; LOAD "L" REGS
; RDDATA ; RDDATA
; -1000 ; -1000 WORDS
; Ibuff ; Ibuff IS Ibuff ADDRESS
; .BYTE 25 ; SECTOR 25
; .BYTE 0 ; TRACK 0
; 312 ; CYLINDER 312
;
; TLOADRK ; LOAD RK REGS
; TWAT32 ; WAIT FOR INTERRUPT
; ERROR 2 ; TO SLOW/NOT COMPLETE ERROR
;
; TCHKOP ; CHECK OPERATION FOR ANY ERRORS
; ERROR 4 ; OR 5, 6, 7, 10 ; REPORT ALL ERRORS
;
; JSR R4, GENCOM ; COMPARE DATA
; 100000
; 1000 ; 1000 WORDS
; BR 15 ; NO MISCOMPARES-SKIP
; ERROR 15 ; PRINT FIRST ERROR
;
; MOV ERRLMT, R0 ; GET ERROR LIMIT
; DEC R0 ; DECREMENT COUNT
; BEQ 65$ ; IF ZERO - EXIT
; JSR R4, GENCOM ; CONTINUE DATA COMPARE
; 40000
; BR 65$ ; NO MORE ERRORS - EXIT
; ERROR 16 ; REPORT NEXT ERROR
; BR 64$ ; LOOP
;
; 64$:
; 65$:
; 15$:
; *****
    
```

```

4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292 016624 000004
4293 016626 012737 000062 001262
4294 016634 104416
4295 016636 104C03
4296
4297
4298 016640 004437 037526
4299 016644 000004
4300 016646 001000
4301
4302 016650 004437 037526
4303 016654 002007
4304 016656 001000
4305
4306 016660 004437 033324
4307 016664 000121
4308 016666 177000
4309 016670 056446
4310 016672 025
4311 016673 002
4312 016674 000312
4313
4314 016676 104417
4315 016700 104426
4316 016702 104002
4317
4318 016704 104421
4319 016706 104004
4320
4321 016710 004437 037526
4322 016714 100000
4323 016716 001000
4324 016720 000413
4325 016722 104015
4326
4327 016724 013700 001634
4328 016730 005300
4329 016732 001406
4330 016734 004437 037526
4331 016740 040000
4332 016742 000402
4333 016744 104016
4334 016746 000770
4335 016750
4336 016750
4337

```

```

*TEST 55 MID-TRANSFER SEEK ON READ (PART 2)
*
* ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
* TRACK 2, SECTOR 25. VERIFY THAT CORRECT DATA IS
* READ AND A MID-TRANSFER SEEK DOES OCCUR.
*
* NOTE: MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE
* EXECUTED BEFORE THIS TEST.
*
*****
TST55: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
;
; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
JSR R4,GENCOM ;GENERATE DATA
4 ;PATTERN 4
1000 ;1000 WORDS
;
JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
2007
1000
;
JSR R4,LRLOAD ;LOAD "L" REGS
RDDATA ;RDDATA
-1000 ;-1000 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 25 ;SECTOR 25
.BYTE 2 ;TRACK 2
312 ;CYLINDER 312
;
TLOADRK ;LOAD RK REGS
TWAT64 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
;
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
;
JSR R4,GENCOM ;COMPARE DATA
100000
1000 ;1000 WORDS
BR 15 ;NO MISCOMPARES-SKIP
ERROR 15 ;REPORT FIRST ERROR
;
MOV ERRLMT,R0 ;GET ERROR LIMIT
64$: DEC R0 ;DECREMENT COUNT
BEQ 65$ ;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 65$ ;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 64$ ;LOOP
65$:
15:
;*****

```

```

4338      ;*TEST 56      CYLINDER ADDRESS OVERFLOW (PART 1)
4339      ;*
4340      ;*          ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
4341      ;*          TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS
4342      ;*          OVERFLOW ERROR DOES NOT OCCUR.
4343      ;*
4344      ;*****
4345      TST56: SCOPE
4346      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
4347      TSSINIT      ;;CLEAR SUBSYSTEM
4348      ERROR      3              ;;BAD INIT ERROR
4349
4350      JSR      R4,LRLOAD      ;;LOAD "L" REGS
4351      RDDATA      ;;RDDATA
4352      -400        ;;-400 WORDS
4353      Ibuff      ;;IBUFF IS BUFF ADDRESS
4354      .BYTE      25          ;;SECTOR 25
4355      .BYTE      2           ;;TRACK 2
4356      632        ;;CYLINDER 632
4357
4358      TLOADRK      ;;LOAD RK REGS
4359      TWAT159     ;;WAIT FOR INTERRUPT
4360      ERROR      2           ;;TO SLOW/NOT COMPLETE ERROR
4361
4362      TCHKOP      ;;CHECK OPERATION FOR ANY ERRORS
4363      ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4364      ;*****
4365      ;*TEST 57      CYLINDER ADDRESS OVERFLOW (PART 2)
4366      ;*
4367      ;*          ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,
4368      ;*          TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS
4369      ;*          OVERFLOW ERROR DOES OCCUR.
4370      ;*
4371      ;*****
4372      TST57: SCOPE
4373      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
4374      TSSINIT      ;;CLEAR SUBSYSTEM
4375      ERROR      3              ;;BAD INIT ERROR
4376
4377      JSR      R4,LRLOAD      ;;LOAD "L" REGS
4378      RDDATA      ;;RDDATA
4379      -401        ;;-401 WORDS
4380      Ibuff      ;;IBUFF IS BUFF ADDRESS
4381      .BYTE      25          ;;SECTOR 25
4382      .BYTE      2           ;;TRACK 2
4383      632        ;;CYLINDER 632
4384
4385      TLOADRK      ;;LOAD RK REGS
4386      TWAT159     ;;WAIT FOR INTERRUPT
4387      ERROR      2           ;;TO SLOW/NOT COMPLETE ERROR
4388
4389      TCHKWE      ;;CHECK OPERATION WITH EXPECTED ERROR
4390      COERR      ;;CYLINDER OVERFLOW
4391      0
4392      0
4393      ERROR      4; OR 5,6,7      ;REPORT ANY DISCREPANCIES

```

4394  
4395  
4396  
4397  
4398  
4399  
4400  
4401  
4402  
4403  
4404  
4405  
4406  
4407  
4408  
4409  
4410  
4411  
4412  
4413  
4414  
4415  
4416  
4417  
4418  
4419  
4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449

017066	000004		
017070	012737	000062	001262
017076	012737	017106	001110
017104	005001		
017106			
017106	104416		
017110	104003		
017112	012737	010127	001600
017120	013737	001626	001610
017126	012737	000074	001602
017134	110137	001607	
017140	012737	060446	001604
017146	012737	000312	001614
017154	004437	037526	
017160	001200		
017162	104417		
017164	104434		
017166	104002		
017170	104421		
017172	104004		
017174	104415		
017176	005701		
017200	001002		
017202	005201		
017204	000740		
017206	012737	017216	001110
017214	005001		
017216			
017216	104416		
017220	104003		
017222	012737	010125	001600
017230	013737	001626	001610
017236	110137	001607	
017242	012737	000312	001614
017250	004437	033644	
017254	104421		
017256	104004		

.SBTTL \*\*18 BIT DATA TRANSFER TESTS

```

*****
*TEST 60          FORMAT IN 24 SECTOR FORMAT
*
*          FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR
*          FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT
*          OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER
*          AFTER READ HEADER.
*****
TST60:  SCOPE
        MOV     #50,$TIMES      ;;DO 50. ITERATIONS
        MOV     #1$,$LPERR     ;;SET LOCAL LOOP ON ERROR
        CLR     R1              ;;CLEAR R1 FOR TRACK COUNTER
1$:
        TSSINIT                ;;CLEAR SUBSYSTEM
        ERROR  3                ;;BAD INIT ERROR
        MOV     #WRHEAD!CFMT,L.CS1 ;;SET UP FOR WRITE HEADER 24(8) SECTOR MODE
        MOV     DRVNUM,L.CS2    ;;SET DRIVE NUMBER
        MOV     #74,L.WC        ;;SET WORD COUNT
        MOV     R1,L.DT         ;;LOAD TRACK ADDRESS
        MOV     #0BUFF,L.BA     ;;      SET BUS ADDRESS
        MOV     #312,L.DCYL     ;;      CYLINDER ADDRESS
        JSR     R4,GENCOM       ;;GENERATE HEADER
        1200                    ;;INCLUDE BAD SECTOR BITS
        TLOADRK                ;;LOAD RK REGS
        TWAT159                ;;WAIT FOR INTERRUPT
        ERROR  2                ;;TO SLOW/NOT COMPLETE ERROR
        TCHKOP                ;;CHECK OPERATION FOR ANY ERRORS
        ERROR  4 ;OR 5, 6, 7    ;;REPORT ALL ERRORS
        SCOP1                    ;;LOCAL LOOP ON ERROR TO 1$
        TST     R1              ;;R1 POINTING TO TRACK 0
        BNE    2$              ;;NO-SKIP
        INC    R1              ;;BUMP TO TRACK 1
        BR     1$              ;;LOOP
        MOV     #3$,$LPERR     ;;SET LOCAL LOOP ON ERROR
        CLR    R1              ;;CLEAR TRACK POINTER
3$:
        TSSINIT                ;;CLEAR SUBSYSTEM
        ERROR  3                ;;BAD INIT ERROR
        MOV     #RDHEAD!CFMT,L.CS1 ;;LOAD READ 24(8) SECTOR FORMAT
        MOV     DRVNUM,L.CS2    ;;LOAD DRIVE NUMBER
        MOV     R1,L.DT         ;;LOAD TRACK
        MOV     #312,L.DCYL     ;;LOAD CYLINDER
        JSR     R4,RDSTHD       ;;GO READ STANDARD HEADER
        TCHKOP                ;;RETURN IF CERR W/O DATA LATE SET
        ERROR  4; OR 5,6,7     ;;REPORT ALL OTHER ERRORS
    
```



```

4450 017260 104013          ERROR 13          ;REPORT DATA LATE
4451 017262 104002          ERROR 2           ;REPORT "OPERATION TO SLOW" OR "HEADER
4452                                     ;D NOT FOUND" MESSAGE
4453
4454 017264 104415          SCOP1           ;LOCAL LOOP TO 3$ ON ERROR
4455 017266 004437 037526   JSR            R4,GENCOM ;GENERATE & COMPARE HEADERS
4456 017272 101200          101200         ;INCLUDING BAD SECTOR LISTS
4457 017274 000413          BR            4$    ;NO MISCOMPARES-SKIP
4458 017276 104015          ERROR 15         ;REPORT FIRST MISCOMPARE
4459
4460 017300 013700 001634   MOV            ERRLMT,R0 ;GET ERROR LIMIT
4461 017304 005300          DEC            R0     ;DECREMENT COUNT
4462 017306 001406          SEQ            65$    ;IF ZERO - EXIT
4463 017310 004437 037526   JSR            R4,GENCOM ;CONTINUE DATA COMPARE
4464 017314 040000          40000
4465 017316 000402          BR            65$    ;NO MORE ERRORS - EXIT
4466 017320 104016          ERROR 16         ;REPORT NEXT ERROR
4467 017322 000770          BR            64$    ;LOOP
4468 017324
4469
4470 017324 104415          4$: SCOP1           ;LOCAL LOOP TO 3$
4471 017326 005701          TST            R1     ;POINTING TO TRACK 1
4472 017330 001002          BNE            5$    ;NO-EXIT
4473 017332 005201          INC            R1     ;BUMP TO TRACK 1
4474 017334 000730          BR            3$    ;LOOP
4475
4476 017336
4477
4478
4479
4480
4481
4482
4483
4484
4485 017336 000004          *****
4486 017340 012737 000062 001262 ;TEST 61      24 SECTOR FORMAT DATA TRANSFER (PART 1)
4487 017346 104416          *****
4488 017350 104003          ;
4489                                     ; ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
4490                                     ; TO CYLINDER 312, TRACK 0, SECTOR 0. READ SECTOR BACK
4491                                     ; AND MAKE SURE IT IS CORRECT.
4492                                     ;
4493                                     ; *****
4494
4495 017352 004437 037526   JSR            R4,GENCOM ;GENERATE DATA
4496 017356 000013          13             ;PATTERN 13
4497 017360 000400          400           ;400 WORDS
4498
4499 017362 004437 037526   JSR            R4,GENCOM ;CLEAR Ibuff TO ALL ONES
4500 017366 002007          2007
4501 017370 000400          400
4502
4503 017372 004437 033324   JSR            R4,LRLOAD ;LOAD "L" REGS
4504 017376 010123          WRDATA:CFMT    ;WRDATA:CFMT
4505 017400 177400          -400          ;-400 WORDS
4506 017402 060446          OBUFF         ;OBUFF IS BUFF ADDRESS
4507 017404 000          .BYTE 0       ;SECTOR 0
4508 017405 000          .BYTE 0       ;TRACK 0
4509 017406 000312          312          ;CYLINDER 312

```



```

4506 017410 104417 TLOADRK ;LOAD RK REGS
4507 017412 104424 TWAT32 ;WAIT FOR INTERRUPT
4508 017414 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4509
4510 017416 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4511 017420 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4512
4513 017422 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
4514 017426 010121 RDDATA:CFMT ;RDDATA:CFMT
4515 017430 177400 -400 ;-400 WORDS
4516 017432 056446 Ibuff ;IBUFF IS BUFF ADDRESS
4517 017434 000 .BYTE 0 ;SECTOR 0
4518 017435 000 .BYTE 0 ;TRACK 0
4519 017436 000312 312 ;CYLINDER 312
4520
4521 017440 104417 TLOADRK ;LOAD RK REGS
4522 017442 104424 TWAT32 ;WAIT FOR INTERRUPT
4523 017444 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4524
4525 017446 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4526 017450 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4527
4528 017452 004437 037526 JSR R4,GENCOM ;COMPARE DATA
4529 017456 100000 100000
4530 017460 000400 400 ;400 WORDS
4531 017462 000413 BR 15 ;NO MISCOMPARES-SKIP
4532 017464 104015 ERROR 15 ;REPORT 1ST ERROR
4533
4534 017466 013700 001634 MOV ERRLMT,RD ;GET ERROR LIMIT
4535 017472 005300 645: DEC RD ;DECREMENT COUNT
4536 017474 001406 BEQ 655 ;IF ZERO - EXIT
4537 017476 004437 037526 JSR R4,GENCOM ;CONTINUE DATA COMPARE
4538 017502 040000 40000
4539 017504 000402 BR 655 ;NO MORE ERRORS - EXIT
4540 017506 104016 ERROR 16 ;REPORT NEXT ERROR
4541 017510 000770 BR 645 ;LOOP
4542 017512 655:
4543
4544 017512 15:
4545 *****
4546 *TEST 62 24 SECTOR FORMAT DATA TRANSFER (PART 2)
4547 *
4548 * LOAD A LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF
4549 * 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0,
4550 * SECTOR 0 WITH BUFFER BEGINNING 110 WORDS BEFORE WORD
4551 * WITH BAD PARITY. MAKE SURE UNIBUS PARITY ERROR DOES NOT SET.
4552 * READ SECTOR BACK AND MAKE SURE IT IS CORRECT.
4553 *
4554 * NOTE: THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
4555 * EXISTS FOR SPECIFIED LOCATION.
4556 *
4557 *****
4558 017512 000004 TST62: SCOPE
4559 017514 012737 000062 001262 MOV #50,STIMES ;DO 50. ITERATIONS
4560 017522 032737 000200 001656 BIT #PARBK0,OPTFLG ;PARITY OPTION PRESENT?
4561 017530 001504 BEQ 45 ;YES-SKIP

```

```

4562
4563 017532 004437 037526 1$: JSR R4,GENCOM ;GENERATE DATA
4564 017536 000014 14 ;PATTERN 14
4565 017540 000400 400 ;400 WORDS
4566
4567 017542 012746 000340 MOV #PR7,-(SP) ;SET PRIORITY TO 7
4568 017546 012746 017554 MOV #10$,-(SP) ;SET ADDRESS
4569 017552 000002 RTI
4570 017554 10$:
4571
4572 017554 032737 000100 001656 BIT #PARBK1,OPTFLG ;TEST IF BANK 1 AVAIL
4573 017562 001403 BEQ 5$ ;NO - SKIP
4574 017564 012777 000004 162110 MOV #BIT2,MMCSR2 ;SET WRONG PARITY WRITE
4575 017572 012777 000004 162100 5$: MOV #BIT2,MMCSR1 ;SET WRONG PARITY BIT
4576 017600 012737 177777 060670 MOV #-1,OBUF+222 ;WRITE WITH BAD PARITY
4577 017606 012737 177777 060666 MOV #-1,OBUF+220
4578 017614 012777 000001 162056 MOV #BIT0,MMCSR1 ;CLEAR WRONG PARITY, SET IE
4579 017622 032737 000100 001656 BIT #PARBK1,OPTFLG ;TEST IF BANK 1 AVAIL
4580 017630 001403 BEQ 6$ ;NO - SKIP
4581 017632 012777 000001 162042 MOV #BIT0,MMCSR2
4582
4583 017640 013746 001622 6$: MOV RKPRI,-(SP) ;RESTORE PRIORITY
4584 017644 012746 017652 MOV #11$,-(SP)
4585 017650 000002 RTI
4586 017652 11$:
4587 017652 104416 TSSINIT ;CLEAR SUBSYSTEM
4588 017654 104003 ERROR 3 ;BAD INIT ERROR
4589 017656 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
4590 017662 010123 WRDATA:CFMT ;WRDATA:CFMT
4591 017664 177400 -400 ;-400 WORDS
4592 017666 060446 OBUF ;OBUF IS BUFF ADDRESS
4593 017670 000 .BYTE 0 ;SECTOR 0
4594 017671 000 .BYTE 0 ;TRACK 0
4595 017672 000312 312 ;CYLINDER 312
4596
4597 017674 104417 TLOADRK ;LOAD RK REGS
4598 017676 104424 TWAT32 ;WAIT FOR INTERRUPT
4599 017700 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4600
4601 017702 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4602 017704 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4603 ; A UNIBUS PARITY ERROR REPORTED AT THIS
4604 ; TIME INDICATES CONTROLLER IS NOT
4605 ; PROPERLY TRANSFERRING DATA IN 18 BIT MODE
4606
4607 017706 005077 161766 CLR MMCSR1 ;CLEAR IE
4608 017712 032737 000100 001656 BIT #PARBK1,OPTFLG
4609 017720 001402 BEQ 7$ 7$
4610 017722 005077 161754 CLR MMCSR2
4611 017726 005037 060670 CLR OBUF+222
4612 017732 005037 060666 CLR OBUF+220
4613 017736 004737 032434 JSR PC,OPTTST ;RESET OPTIONS
4614
4615 017742 4$:
4616 ;*****
4617 ;*TEST 63 24 SECTOR FORMAT DATA TRANSFER (PART 3)

```

# E10

4618  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626  
4627  
4628  
4629  
4630  
4631  
4632  
4633  
4634  
4635  
4636  
4637  
4638  
4639  
4640  
4641  
4642  
4643  
4644  
4645  
4646  
4647  
4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657  
4658  
4659  
4660  
4661  
4662  
4663  
4664  
4665  
4666  
4667  
4668  
4669  
4670  
4671  
4672  
4673

017742 000004  
017744 012737 000062 001262  
017752 004737 032434  
017756 104416  
017760 104003  
  
017762 004437 037526  
017766 000015  
017770 001000  
  
017772 004437 037526  
017776 002007  
020000 001000  
  
020002 004437 033324  
020006 010123  
020010 177000  
020012 060446  
020014 023  
020015 000  
020016 000312  
  
020020 104417  
020022 104426  
020024 104002  
  
020026 104421  
020030 104004  
  
020032 004437 033324  
020036 010121  
020040 177000  
020042 056446  
020044 023  
020045 000  
020046 000312  
  
020050 104417  
020052 104426  
020054 104002  
  
020056 104421  
020060 104004  
  
020062 004437 037526  
020066 100000  
020070 001000  
020072 000413  
020074 104015

```

:*
:*      ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT
:*      TO CYLINDER 312, TRACK 0, SECTOR 23.  READ SECTOR BACK
:*      AND MAKE SURE IT IS CORRECT.  MAKE SURE THAT MID-TRANSFER
:*      SEEK HAS TAKEN PLACE.
:*
:*****
:ST63: SCOPE
MOV      #50, $TIMES      ;; DO 50. ITERATIONS
JSR      PC, OPTTST      ;; SET UP OPTIONS
TSSINIT      ;; CLEAR SUBSYSTEM
ERROR    3                ;; BAD INIT ERROR

JSR      R4, GENCOM      ;; GENERATE DATA
15       ;; PATTERN 15
1000     ;; 1000 WORDS

JSR      R4, GENCOM      ;; CLEAR Ibuff TO ALL ONES
2007     ;;
1000     ;;

JSR      R4, LRLOAD      ;; LOAD "L" REGS
WRDATA!CFMT      ;; WRDATA!CFMT
-1000     ;; -1000 WORDS
OBUFF     ;; OBUFF IS BUFF ADDRESS
.BYTE    23              ;; SECTOR 23
.BYTE    0                ;; TRACK 0
312      ;; CYLINDER 312

TLOADRK      ;; LOAD RK REGS
TWAT64      ;; WAIT FOR INTERRUPT
ERROR    2                ;; TO SLOW/NOT COMPLETE ERROR

TCHKOP      ;; CHECK OPERATION FOR ANY ERRORS
ERROR    4 ;OR 5, 6, 7, 10 ;; REPORT ALL ERRORS

JSR      R4, LRLOAD      ;; LOAD "L" REGS
RDATA!CFMT      ;; RDATA!CFMT
-1000     ;; -1000 WORDS
IBUFF     ;; IBUFF IS BUFF ADDRESS
.BYTE    23              ;; SECTOR 23
.BYTE    0                ;; TRACK 0
312      ;; CYLINDER 312

TLOADRK      ;; LOAD RK REGS
TWAT64      ;; WAIT FOR INTERRUPT
ERROR    2                ;; TO SLOW/NOT COMPLETE ERROR

TCHKOP      ;; CHECK OPERATION FOR ANY ERRORS
ERROR    4 ;OR 5, 6, 7, 10 ;; REPORT ALL ERRORS

JSR      R4, GENCOM      ;; COMPARE DATA
100000     ;;
1000       ;; 1000 WORDS
BR        15              ;; NO MISCOMPARES-SKIP
ERROR    15              ;; REPORT FIRST ERROR
  
```

4674								
4675	020076	013700	001634					
4676	020102	005300		64\$:	MOV	ERRLMT,RO		;GET ERROR LIMIT
4677	020104	001406			DEC	RO		;DECREMENT COUNT
4678	020106	004437	037526		BEG	65\$		;IF ZERO - EXIT
4679	020112	040000			JSR	R4,GENCOM		;CONTINUE DATA COMPARE
4680	020114	000402			40000			
4681	020116	104016			BR	65\$		;NO MORE ERRORS - EXIT
4682	020120	000770			ERROR	16		;REPORT NEXT ERROR
4683	020122			65\$:	BR	64\$		;LOOP
4684								
4685	020122			1\$:				
4686								
4687								
4688								
4689								
4690								
4691								
4692								
4693								
4694								
4695								
4696								
4697								
4698								
4699								
4700								
4701								
4702								
4703	020122	000004						
4704	020124	012737	000062 001262					
4705	020132	104416						
4706	020134	104003						
4707								
4708	020136	004437	033324					
4709	020142	000127						
4710	020144	177676						
4711	020146	060446						
4712	020150	000						
4713	020151	000						
4714	020152	000312						
4715								
4716	020154	004437	037526					
4717	020160	000600						
4718								
4719	020162	042737	040000 060456					
4720	020170	042737	040000 060460					
4721								
4722	020176	104417						
4723	020200	104431						
4724	020202	104002						
4725								
4726	020204	104421						
4727	020206	104004						
4728								
4729	020210	004437	037526					

```

        MOV     ERRLMT,RO      ;GET ERROR LIMIT
        DEC     RO            ;DECREMENT COUNT
        BEQ    65$           ;IF ZERO - EXIT
        JSR    R4,GENCOM     ;CONTINUE DATA COMPARE
        40000
        BR     65$           ;NO MORE ERRORS - EXIT
        ERROR  16            ;REPORT NEXT ERROR
        BR     64$           ;LOOP

        65$:
        1$:

        .SBTTL **SPECIAL DATA TRANSFER TESTS

        ;*****
        ;*TEST 64      MULTI-SECTOR DATA TRANSFER AND BSE
        ;*
        ;*      FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH
        ;*      SECTOR 1 MARKED BAD.  ISSUE A WRITE DATA OF 1000 WORDS
        ;*      TO CYLINDER 312, TRACK 0, SECTOR 0.  MAKE SURE BAD SECTOR
        ;*      ERROR SETS AND RKDA IS CORRECT.  READ SECTOR 0 AND
        ;*      MAKE SURE IT IS CORRECT.
        ;*
        ;*      ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0,
        ;*      SECTOR 0.  MAKE SURE BAD SECTOR ERROR SETS AND THE
        ;*      PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.
        ;*
        ;*****
        ;*TEST64:  SCOPE
        ;*      MOV     #50.,$TIMES      ;DO 50. ITERATIONS
        ;*      TSSINIT      ;CLEAR SUBSYSTEM
        ;*      ERROR  3      ;BAD INIT ERROR
        ;*
        ;*      JSR    R4,LRLOAD      ;LOAD "L" REGS
        ;*      WRHEAD      ;WRHEAD
        ;*      -102      ;-102 WORDS
        ;*      OBUFF      ;OBUFF IS BUFF ADDRESS
        ;*      .BYTE  0      ;SECTOR 0
        ;*      .BYTE  0      ;TRACK 0
        ;*      312      ;CYLINDER 312
        ;*
        ;*      JSR    R4,GENCOM      ;BUILD HEADERS
        ;*      600
        ;*
        ;*      BIC    #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD
        ;*      BIC    #BIT14,OBUFF+12 ;CORRECT HURC
        ;*
        ;*      TLOADRK      ;LOAD RK REGS
        ;*      TWAT112      ;WAIT FOR INTERRUPT
        ;*      ERROR  2      ;TO SLOW/NOT COMPLETE ERROR
        ;*
        ;*      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
        ;*      ERROR  4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
        ;*
        ;*      JSR    R4,GENCOM      ;GENERATE DATA
    
```

```

4730 020214 000016          16          ;PATTERN 16
4731 020216 001000          1000         ;1000 WORDS
4732
4733 020220 004437 037526    JSR      R4,GENCOM    ;CLEAR IBUFF TO ALL ONES
4734 020224 002007          2007
4735 020226 001000          1000
4736
4737 020230 004437 033324    JSR      R4,LRLOAD    ;LOAD "L" REGS
4738 020234 000123          WRDATA       ;WRDATA
4739 020236 177000          -1000        ;-1000 WORDS
4740 020240 060446          OBUFF        ;OBUFF IS BUFF ADDRESS
4741 020242          000          .BYTE 0        ;SECTOR 0
4742 020243          000          .BYTE 0        ;TRACK 0
4743 020244 000312          312         ;CYLINDER 312
4744
4745 020246 104417          TLOADRK      ;LOAD RK REGS
4746 020250 104424          TWAT32       ;WAIT FOR INTERRUPT
4747 020252 104002          ERROR 2     ;TO SLOW/NOT COMPLETE ERROR
4748
4749 020254 104422          TCHKWE       ;CHECK OPERATION WITH EXPECTED ERR
4750 020256 000000          0
4751 020260 000100          BSERR        ;BAD SECTOR ERROR
4752 020262 000000          0
4753 020264 104004          ERROR 4: OR 5,6,7 ;REPORT ALL DISCREPANCIES
4754 020266 005037 044756    CLR      GRP4ER     ;CLEAR GROUP 4 ERRORS
4755 020272 004437 035116    JSR      R4,CHKCTS  ;CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
4756 020276 032737 000020 044756    BIT      #TRKERR,GRP4ER ;TRK IN ERROR?
4757 020304 001416          BEQ      1$        ;NO-SKIP
4758 020306 012737 050732 001450    MOV      #EM13,EM10N ;"TRACK ADDRESS INCORRECT"
4759 020314 013737 044732 001202    MOV      EXPTRK,$REG10 ;EXPECTED VALUE
4760 020322 013737 044744 001204    MOV      REALTRK,$REG11 ;REAL VALUE
4761 020330 012737 045443 055524    MOV      #OPER37,DF010A ;"AFTER WRITE DATA TERMINATED WITH BSE"
4762 020336 104010          ERROR 10
4763 020340 000527          BR      5$        ;EXIT
4764
4765 020342 032737 000040 044756 1$:    BIT      #SECERR,GRP4ER ;SECTOR IN ERROR?
4766 020350 001422          BEQ      3$        ;NO-SKIP
4767 020352 012737 050762 001450    MOV      #EM14,EM10N ;"SECTOR ADDRESS INCORRECT"
4768 020360 012737 045443 055524    MOV      #OPER37,DF010A ;"AFTER WRITE DATA ABORTED WITH BSE"
4769 020366 013737 044730 001202    MOV      EXPSEC,$REG10 ;EXPECTED VALUE
4770 020374 013737 044746 001204    MOV      REALSEC,$REG11 ;REAL VALUE
4771 020402 104010          ERROR 10
4772 020404 000505          BR      5$        ;EXIT
4773 020406 104415          SCOP1
4774 020410 012737 020416 001110    MOV      #3$,SLPERR  ;LOCAL LOOP TO BEGINNING OF TEST
4775 020416          3$:
4776 020416 104416          TSSINIT      ;CLEAR SUBSYSTEM
4777 020420 104003          ERROR 3        ;BAD INIT ERROR
4778 020422 004437 033324    JSR      R4,LRLOAD  ;LOAD "L" REGS
4779 020426 000121          RDDATA       ;RDDATA
4780 020430 177400          -400         ;-400 WORDS
4781 020432 056446          IBUFF        ;IBUFF IS BUFF ADDRESS
4782 020434          000          .BYTE 0        ;SECTOR 0
4783 020435          000          .BYTE 0        ;TRACK 0
4784 020436 000312          312         ;CYLINDER 312
4785

```

4786	020440	104417		TLOADRK		;LOAD RK REGS
4787	020442	104424		TWAT32		;WAIT FOR INTERRUPT
4788	020444	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
4789						
4790	020446	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
4791	020450	104004		ERROR 4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
4792						
4793	020452	004437	037526	JSR R4,GENCOM		;COMPARE DATA
4794	020456	100000		100000		
4795	020460	000400		400		;400 WORDS
4796	020462	000413		BR 4\$		;NO MISCOMPARES-EXIT
4797	020464	104015		ERROR 15		;REPORT FIRST ERROR
4798						
4799	020466	013700	001634	MOV ERRLMT,RO		;GET ERROR LIMIT
4800	020472	005300		64\$: DEC RO		;DECREMENT COUNT
4801	020474	001406		BEQ 65\$		;IF ZERO - EXIT
4802	020476	004437	037526	JSR R4,GENCOM		;CONTINUE DATA COMPARE
4803	020502	040000		40000		
4804	020504	000402		BR 65\$		;NO MORE ERRORS - EXIT
4805	020506	104016		ERROR 16		;REPORT NEXT ERROR
4806	020510	000770		BR 64\$		;LOOP
4807	020512			65\$:		
4808						
4809	020512	004437	037526	4\$: JSR R4,GENCOM		;CLEAR Ibuff
4810	020516	002007		2007		
4811	020520	001000		1000		
4812						
4813	020522	004437	033324	JSR R4,LRLOAD		;LOAD "L" REGS
4814	020526	000121		RDDATA		;RDDATA
4815	020530	177000		-1000		; -1000 WORDS
4816	020532	056446		IBUFF		;IBUFF IS BUFF ADDRESS
4817	020534	000		.BYTE 0		;SECTOR 0
4818	020535	000		.BYTE 0		;TRACK 0
4819	020536	000312		312		;CYLINDER 312
4820						
4821	020540	104417		TLOADRK		;LOAD RK REGS
4822	020542	104424		TWAT32		;WAIT FOR INTERRUPT
4823	020544	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
4824						
4825	020546	104422		TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
4826	020550	000000		0		
4827	020552	000100		BSERR		;BAD SECTOR ERROR
4828	020554	000000		0		
4829	020556	104004		ERROR 4; OR 5,6,7		;REPORT ALL DISCREPANCIES
4830						
4831	020560	004437	037526	JSR R4,GENCOM		;COMPARE DATA AGAIN
4832	020564	100000		100000		
4833	020566	000400		400		;400 WORDS
4834	020570	000413		BR 5\$		;NO MISCOMPARES
4835	020572	104015		ERROR 15		;REPORT FIRST ERROR
4836						
4837	020574	013700	001634	MOV ERRLMT,RO		;GET ERROR LIMIT
4838	020600	005300		66\$: DEC RO		;DECREMENT COUNT
4839	020602	001406		BEQ 67\$		;IF ZERO - EXIT
4840	020604	004437	037526	JSR R4,GENCOM		;CONTINUE DATA COMPARE
4841	020610	040000		40000		

```

4842 020612 000402          BR      67$          ;NO MORE ERRORS - EXIT
4843 020614 104016          ERROR   16          ;REPORT NEXT ERROR
4844 020616 000770          BR      66$          ;LOOP
4845 020620
4846
4847 020620
4848
4849
4850
4851
4852
4853
4854
4855
4856 020620 000004          TST65: SCOPE
4857 020622 012737 000001 001262      MOV     #1,$TIMES      ;;DO 1 ITERATION
4858 020630 005000          CLR     RO            ;CLEAR TRACK COUNTER
4859 020632 012737 020640 001110      MOV     #1$,$LPERR    ;SET LOCAL LOOP
4860
4861 020640
4862 020640 104416          1$:      TSSINIT
4863 020642 104003          ERROR   3            ;CLEAR SUBSYSTEM
4864
4865 020644 013737 001626 001610      MOV     DRVNUM,L,CS2  ;LOAD DRIVE NUMBER
4866 020652 012737 000127 001600      MOV     #WRHEAD,L,CS1 ;LOAD WRITE HEADER
4867 020660 110037 001607 001600      MOV     RO,L,DT       ;LOAD DESIRED TRACK FROM TRACK COUNTER
4868 020664 012737 060446 001604      MOV     #0BUFF,L,BA   ;LOAD BUS ADDRESS
4869 020672 012737 177676 001602      MOV     #-102,L,WC    ;WORD COUNT
4870 020700 012737 000312 001614      MOV     #312,L,DCYL   ;CYLINDER
4871
4872 020706 004437 037526          JSR     R4,GENCOM     ;BUILD HEADER
4873 020712 001200          1200          ;WITH BSE FLAGGED
4874
4875 020714 104417          TLOADRK
4876 020716 104431          TWTAT112
4877 020720 104002          ERROR   2            ;LOAD RK REGS
4878
4879 020722 104421          TCHKOP
4880 020724 104004          ERROR   4 ;OR 5, 6, 7 ;CHECK OPERATION FOR ANY ERRORS
4881
4882 020726 104415          SCOP1            ;REPORT ALL ERRORS
4883
4884 020730 005700          TST     RO            ;LOCAL LOOP TO 1$
4885 020732 001002          BNE    2$          ;RO AT ZERO?
4886 020734 005200          INC    RO            ;NO-EXIT
4887 020736 000740          BR     1$          ;BUMP COUNTER
4888 020740 005000          CLR    RO            ;LOOP
4889 020742 012737 020750 001110      2$:      MOV     #113$,$LPERR  ;CLEAR SECTOR COUNTER
4890
4891 020750          113$:
4892 020750 104416          TSSINIT
4893 020752 104003          ERROR   3            ;SET LOCAL LOOP ON ERROR
4894
4895 020754 004437 033324          3$:      JSR     R4,LRLOAD
4896 020760 000121          RDDATA
4897 020762 177400          -400          ;LOAD "L" REGS
                    ;RDDATA
                    ;-400 WORDS

```

```

4898 020764 056446          Ibuff          ;IBUFF IS BUFF ADDRESS
4899 020766      000          .BYTE 0          ;SECTOR 0
4900 020767      000          .BYTE 0          ;TRACK 0
4901 020770 000312          312          ;CYLINDER 312
4902
4903 020772 110037 001606    MOVb          RD,L,DS ;LOAD SECTOR COUNTER INTO DESIRED SECTOR
4904
4905 020776 104417          TLOADRK        ;LOAD RK REGS
4906 021000 104424          TWAT32        ;WAIT FOR INTERRUPT
4907 021002 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
4908
4909 021004 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
4910 021006 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4911
4912 021010 104415          SCOPE1        ;LOCAL LOOP TO 3$ ON ERROR
4913
4914 021012 022700 000024    CMP #24,RD     ;LAST SECTOR READ?
4915 021016 001402          BEQ 4$        ;YES-EXIT
4916 021020 005200          INC RD        ;BUMP SECTOR COUNTER
4917 021022 000754          BR 3$        ;LOOP
4918
4919 021024          4$:

```

4\$:

.SBTTL \*\*WRITE CHECK TESTS

```

*****
: *TEST 66      WRITE-CHECK WITH NO ERROR
: *
: *      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
: *      DO A WRITE-CHECK OF 400 WORDS.  MAKE SURE NO
: *      ERROR OCCURS.
: *
*****

```

```

TST66: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
4936 021040 004437 033324    JSR R4,LRLOAD ;LOAD "L" REGS
4937 021044 000123          WRDATA        ;WRDATA
4938 021046 177400          -400          ;-400 WORDS
4939 021050 060446          OBUFF         ;OBUFF IS BUFF ADDRESS
4940 021052      000          .BYTE 0          ;SECTOR 0
4941 021053      000          .BYTE 0          ;TRACK 0
4942 021054 000312          312          ;CYLINDER 312
4943
4944 021056 004437 037526    JSR R4,GENCOM ;GENERATE DATA
4945 021062 000002          2            ;PATTERN 2
4946 021064 000400          400          ;400 WORDS
4947
4948 021066 104417          TLOADRK        ;LOAD RK REGS
4949 021070 104424          TWAT32        ;WAIT FOR INTERRUPT
4950 021072 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
4951
4952 021074 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
4953 021076 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

```



# K10

```
4954
4955 021100 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
4956 021104 000131 WRTCHK ;WRTCHK
4957 021106 177400 -400 ;-400 WORDS
4958 021110 060446 OBUFF ;OBUFF IS BUFF ADDRESS
4959 021112 000 .BYTE 0 ;SECTOR 0
4960 021113 000 .BYTE 0 ;TRACK 0
4961 021114 000312 312 ;CYLINDER 312
4962
4963 021116 104417 TLOADRK ;LOAD RK REGS
4964 021120 104424 TWAT32 ;WAIT FOR INTERRUPT
4965 021122 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4966
4967 021124 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4968 021126 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4969
4970 *****
4971 *TEST 67 WRITE CHECK ERROR (PART 1)
4972 *
4973 * WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
4974 * WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME.
4975 * DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
4976 * CONFIGURATIONS:
4977 *
4978 * 000001 000020 000400 010000
4979 * 000002 000040 001000 020000
4980 * 000004 000100 002000 040000
4981 * 000010 000200 004000 100000
4982 *
4983 * MAKE SURE WRITE CHECK ERROR SET FOR EACH
4984 * OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
4985 * AND WORD COUNT IS CORRECT.
4986 *
4987 *****
4988 *TST67: SCOPE
4989 MOV #50, $TIMES ;DO 50. ITERATIONS
4990 MOV #BIT0,RO ;SET LO ORDER BIT IN RO FOR
;CAUSING WRITE CHECK ERROR
4991 021144 104416 TSSINIT ;CLEAR SUBSYSTEM
4992 021146 104003 ERROR 3 ;BAD INIT ERROR
4993 021150 004437 037526 JSR R4,GENCOM ;GENERATE DATA, ALL 0'S
4994 021154 000001 1
4995 021156 000400 400
4996
4997 021160 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
4998 021164 000123 WRDATA ;WRDATA
4999 021166 177400 -400 ;-400 WORDS
5000 021170 060446 OBUFF ;OBUFF IS BUFF ADDRESS
5001 021172 000 .BYTE 0 ;SECTOR 0
5002 021173 000 .BYTE 0 ;TRACK 0
5003 021174 000312 312 ;CYLINDER 312
5004
5005 021176 104417 TLOADRK ;LOAD RK REGS
5006 021200 104424 TWAT32 ;WAIT FOR INTERRUPT
5007 021202 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5008
5009 021204 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
```

```

5010 021206 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5011
5012 021210 004437 033324  JSR    R4,LRLOAD      ;LOAD "L" REGS
5013 021214 000131          WRTCHK          ;WRTCHK
5014 021216 177400          -400           ; -400 WORDS
5015 021220 060446          OBUFF          ;OBUFF IS BUFF ADDRESS
5016 021222      000        .BYTE 0         ;SECTOR 0
5017 021223      000        .BYTE 0         ;TRACK 0
5018 021224 000312          312           ;CYLINDER 312
5019
5020 021226 104417          TLOADRK        ;LOAD RK REGS
5021 021230 104424          TWAT32         ;WAIT FOR INTERRUPT
5022 021232 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
5023
5024 021234 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5025 021236 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5026
5027 021240 104415          SCOPI         ;LOCAL LOOP ON WRITE CHECK
5028 021242 012737 021250 001110  MOV    #1$, $LPERR ;SET LOCAL LOOP
5029 021250 010037 060666      1$:      MOV    R0, OBUFF+220 ;CAUSE ERROR BIT IN BUFFER
5030 021254 104416          TSSINIT       ;CLEAR SUBSYSTEM
5031 021256 104003          ERROR 3        ;BAD INIT ERROR
5032 021260 004437 033324  JSR    R4,LRLOAD ;LOAD "L" REGS
5033 021264 000131          WRTCHK          ;WRTCHK
5034 021266 177400          -400           ; -400 WORDS
5035 021270 060446          OBUFF          ;OBUFF IS BUFF ADDRESS
5036 021272      000        .BYTE 0         ;SECTOR 0
5037 021273      000        .BYTE 0         ;TRACK 0
5038 021274 000312          312           ;CYLINDER 312
5039
5040 021276 104417          TLOADRK        ;LOAD RK REGS
5041 021300 104424          TWAT32         ;WAIT FOR INTERRUPT
5042 021302 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
5043
5044 021304 104422          TCHKWE        ;CHECK OPERATION WITH EXPECTED ERROR
5045 021306 000000          0
5046 021310 000004          WCKERR        ;WRITE CHECK ERROR
5047 021312 000000          0
5048 021314 104004          ERROR 4; OR 5,6,7 ;REPORT ALL DISCREPANCIES
5049
5050 021316 104415          SCOPI         ;LOCAL LOOP ON ERROR TO 1$
5051
5052      :
5053      :
5054      :
5055      :
5056      :
5057      :
5058      :
5059      :
5060 021320 023727 001544 060670  CMP    T.BA, #OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
5061 021326 001416          BEQ    2$      ;YES-SKIP
5062 021330 101037          BHI    6$      ;IF TO HI - SKIP
5063 021332 012737 050575 001450  MOV    #EM11, EM10N ;"INCORRECT BA"
5064 021340 012737 060670 001202  MOV    #OBUFF+222, $REG10 ;GOOD VALUE
5065 021346 013737 001544 001204  MOV    T.BA, $REG11 ;BAD VALUE
    
```

NOTE: THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF TWO VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE NPR CYCLE, WC AND BA WILL BE ONE PAST WHERE THE ERROR ACTUALLY OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS REFLECT THE ACTUAL WORD WHERE THE ERROR WAS.

```

5066 021354 012737 045477 055524      MOV    #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5067 021362 104010                      ERROR  10
5068
5069 021364 023727 001542 177511 2$:    CMP    T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
5070 021372 001460                      BEQ    3$              ;YES-SKIP
5071 021374 012737 050550 001450      MOV    #EM10,EM10N    ;"INCORRECT WC"
5072 021402 012737 045477 055524      MOV    #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5073 021410 012737 177511 001202      MOV    #-267,$REG10   ;GOOD VALUE
5074 021416 013737 001542 001204      MOV    T.WC,$REG11   ;ERROR VALUE
5075 021424 104010                      ERROR  10
5076 021426 000442                      BR     3$              ;EXIT
5077
5078 021430 023727 001544 060672 6$:    CMP    T.BA,#OBUFF+224 ;TEST IF BA AT HI SIDE
5079 021436 001415                      BEQ    7$              ;YES - SKIP
5080 021440 012737 050575 001450      MOV    #EM11,EM10N   ;SET MESSAGE
5081 021446 012737 060672 001202      MOV    #OBUFF+224,$REG10 ;GOOD VALUE
5082 021454 013737 001544 001204      MOV    T.BA,$REG11   ;ERROR VALUE
5083 021462 012737 045477 055524      MOV    #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5084 021470 104010                      ERROR  10
5085
5086 021472 023727 001542 177512 7$:    CMP    T.WC,#-266     ;TEST IF WORD COUNT AT HI SIDE
5087 021500 001415                      BEQ    3$              ;YES - SKIP
5088 021502 012737 050550 001450      MOV    #EM10,EM10N   ;SET MESSAGE
5089 021510 012737 045477 055524      MOV    #OPER41,DF010A ;"WC ABORTED WITH WCE"
5090 021516 012737 177512 001202      MOV    #-266,$REG10  ;GOOD VALUE
5091 021524 013737 001542 001204      MOV    T.WC,$REG11   ;ERROR VALUE
5092 021532 104010                      ERROR  10
5093
5094 021534 104415                      3$:    SCOP1              ;LOCAL LOOP ON ERROR TO 1$
5095
5096 021536 032700 100000                BIT    #BIT15,RO      ;BIT 15 SET?
5097 021542 001002                      BNE   4$              ;YES-EXIT
5098 021544 006300                      ASL   RO              ;SHIFT ERROR BIT
5099 021546 000640                      BR    1$              ;LOOP

```

```

5100
5101 021550      4$:
5102 *****
5103 *TEST 70      WRITE CHECK ERROR (PART 2)
5104 *
5105 *      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
5106 *      IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0,
5107 *      SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
5108 *      ONE OF THE FOLLWOING CONFIGURATIONS:
5109 *
5110 *      177776 177757 177377 167777
5111 *      177775 177737 176777 157777
5112 *      177773 177677 175777 137777
5113 *      177767 177577 173777 077777
5114 *
5115 *      MAKE SURE WRITE CHECK ERROR SET FOR EACH
5116 *      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
5117 *      AND WORD COUNT IS CORRECT.
5118 *
5119 *****
5120 TST70:  SCOPE
5121 021550 000004      MOV    #50.,$TIMES    ;;DO 50. ITERATIONS
021552 012737 000062 001262

```

```

5122 021560 012700 177776      MOV      #177776,R0      ;LOAD R0 FOR CAUSING WRITE CHECK ERROR
5123
5124 021564 104416      TSSINIT      ;CLEAR SUBSYSTEM
5125 021566 104003      ERROR 3      ;BAD INIT ERROR
5126 021570 004437 037526      JSR      R4,GENCOM    ;GENERATE DATA
5127 021574 000007      7          ;ALL 1'S
5128 021576 000400      400       ;400 WORDS
5129
5130 021600 004437 033324      JSR      R4,LRLOAD    ;LOAD "L" REGS
5131 021604 000123      WRDATA      ;WRDATA
5132 021606 177400      -400       ;-400 WORDS
5133 021610 060446      OBUFF      ;OBUFF IS BUFF ADDRESS
5134 021612      000       ;SECTOR 0
5135 021613      000       ;TRACK 0
5136 021614 000312      312       ;CYLINDER 312
5137
5138 021616 104417      TLOADRK     ;LOAD RK REGS
5139 021620 104424      TWAT32     ;WAIT FOR INTERRUPT
5140 021622 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
5141
5142 021624 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
5143 021626 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5144
5145 021630 004437 033324      JSR      R4,LRLOAD    ;LOAD "L" REGS
5146 021634 000131      WRTCHK     ;WRTCHK
5147 021636 177400      -400       ;-400 WORDS
5148 021640 060446      OBUFF      ;OBUFF IS BUFF ADDRESS
5149 021642      000       ;SECTOR 0
5150 021643      000       ;TRACK 0
5151 021644 000312      312       ;CYLINDER 312
5152
5153 021646 104417      TLOADRK     ;LOAD RK REGS
5154 021650 104424      TWAT32     ;WAIT FOR INTERRUPT
5155 021652 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
5156
5157 021654 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
5158 021656 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5159
5160 021660 104415      SCOP1     ;LOCAL LOOP TO START OF TEST
5161 021662 012737 021670 001110      MOV      #15,$LPERR   ;SET LOCAL LOOP
5162
5163 021670 010037 060666      15:      MOV      R0,OBUFF+220 ;PUT WORD IN OBUFF TO CAUSE WCE
5164 021674 104416      TSSINIT     ;CLEAR SUBSYSTEM
5165 021676 104003      ERROR 3     ;BAD INIT ERROR
5166
5167 021700 004437 033324      JSR      R4,LRLOAD    ;LOAD "L" REGS
5168 021704 000131      WRTCHK     ;WRTCHK
5169 021706 177400      -400       ;-400 WORDS
5170 021710 060446      OBUFF      ;OBUFF IS BUFF ADDRESS
5171 021712      000       ;SECTOR 0
5172 021713      000       ;TRACK 0
5173 021714 000312      312       ;CYLINDER 312
5174
5175 021716 104417      TLOADRK     ;LOAD RK REGS
5176 021720 104424      TWAT32     ;WAIT FOR INTERRUPT
5177 021722 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR

```

```

5178
5179 021724 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR
5180 021726 000000 0
5181 021730 000004 WCKERR ;WRITE CHECK ERROR
5182 021732 000000 0
5183 021734 104004 ERROR 4; OR 5,6,7 ;REPORT ALL DISCREPANCIES
5184
5185 021736 104415 SCOPI ;LOCAL LOOP TO 1$
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195 021740 023727 001544 060670 CMP T.BA,#OBUF+222 ;CHECK BA HALT AT PROPER PLACE
5196 021746 001416 BEQ 2$ ;YES-SKIP
5197 021750 101037 BHI 6$ ;IF TO HI - SKIP
5198 021752 012737 050575 001450 MOV #EM11,EM10N ;"INCORRECT BA"
5199 021760 012737 060670 001202 MOV #OBUF+222,$REG10 ;GOOD VALUE
5200 021766 013737 001544 001204 MOV T.BA,$REG11 ;BAD VALUE
5201 021774 012737 045477 055524 MOV #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5202 022002 104010 ERROR 10
5203
5204 022004 023727 001542 177511 2$: CMP T.WC,#-267 ;CHECK WORD COUNT AT CORRECT VALUE
5205 022012 001460 BEQ 3$ ;YES-SKIP
5206 022014 012737 050550 001450 MOV #EM10,EM10N ;"INCORRECT WC"
5207 022022 012737 045477 055524 MOV #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5208 022030 012737 177511 001202 MOV #-267,$REG10 ;GOOD VALUE
5209 022036 013737 001542 001204 MOV T.WC,$REG11 ;ERROR VALUE
5210 022044 104010 ERROR 10
5211 022046 000442 BR 3$ ;EXIT
5212
5213 022050 023727 001544 060672 6$: CMP T.BA,#OBUF+224 ;TEST IF BA AT HI SIDE
5214 022056 001415 BEQ 7$ ;YES - SKIP
5215 022060 012737 050575 001450 MOV #EM11,EM10N ;SET MESSAGE
5216 022066 012737 060672 001202 MOV #OBUF+224,$REG10 ;GOOD VALUE
5217 022074 013737 001544 001204 MOV T.BA,$REG11 ;ERROR VALUE
5218 022102 012737 045477 055524 MOV #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
5219 022110 104010 ERROR 10
5220
5221 022112 023727 001542 177512 7$: CMP T.WC,#-266 ;TEST IF WORD COUNT AT HI SIDE
5222 022120 001415 BEQ 3$ ;YES - SKIP
5223 022122 012737 050550 001450 MOV #EM10,EM10N ;SET MESSAGE
5224 022130 012737 045477 055524 MOV #OPER41,DF010A ;"WC ABORTED WITH WCE"
5225 022136 012737 177512 001202 MOV #-266,$REG10 ;GOOD VALUE
5226 022144 013737 001542 001204 MOV T.WC,$REG11 ;ERROR VALUE
5227 022152 104010 ERROR 10
5228
5229 022154 104415 3$: SCOPI
5230
5231 022156 032700 100000 BIT #BIT15,RO ;BIT 15 SET? (ALL PATTERNS TESTED)
5232 022162 001002 BNE 4$ ;YES-EXIT
5233 022164 006300 ASL RO ;SHIFT FOR NEXT TEST
    
```

NOTE: THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF TWO VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE NPR CYCLE, WC AND BA WILL BE ONE PAST WHERE THE ERROR ACTUALLY OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS REFLECT THE ACTUAL WORD WHERE THE ERROR WAS.

```

5234 022166 000640          BR      IS          ;LOOP
5235
5236 022170          45:
5237          ;*****
5238          ;TEST 71      WRITE CHECK OF PARTIAL SECTOR
5239          ;
5240          ;WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
5241          ;CONFIGURATIONS.  ISSUE A WRITE CHECK COMMAND OF
5242          ;110 WORDS MAKING SURE THE 111TH WORD IS
5243          ;DIFFERENT THAN DATA ON DISK.  MAKE SURE
5244          ;WRITE CHECK ERROR DOES NOT SET.
5245          ;
5246          ;*****
5247 022170 000004          TST71: SCOPE
5248 022172 012737 000062 001262  MOV     #50.,$TIMES      ;;DO 50. ITERATIONS
5249 022200 104416          TSSINIT          ;CLEAR SUBSYSTEM
5250 022202 104003          ERROR    3           ;BAD INIT ERROR
5251
5252 022204 004437 037526    JSR     R4,GENCOM      ;GENERATE DATA
5253 022210 000007          7              ;ALL 1'S
5254 022212 000400          400            ;400 WORDS
5255
5256 022214 004437 033324    JSR     R4,LRLoad     ;LOAD "L" REGS
5257 022220 000123          WRDATA          ;WRDATA
5258 022222 177400          -400           ;-400 WORDS
5259 022224 060446          OBUFF          ;OBUFF IS BUFF ADDRESS
5260 022226 000          .BYTE    0        ;SECTOR 0
5261 022227 000          .BYTE    0        ;TRACK 0
5262 022230 000312          312           ;CYLINDER 312
5263
5264 022232 104417          TLOADRK        ;LOAD RK REGS
5265 022234 104424          TWAT32        ;WAIT FOR INTERRUPT
5266 022236 104002          ERROR    2      ;TO SLOW/NOT COMPLETE ERROR
5267
5268 022240 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5269 022242 104004          ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5270
5271 022244 005037 060670    CLR     OBUFF+222
5272
5273 022250 004437 033324    JSR     R4,LRLoad     ;LOAD "L" REGS
5274 022254 000131          WRTCHK        ;WRTCHK
5275 022256 177670          -110          ;-110 WORDS
5276 022260 060446          OBUFF          ;OBUFF IS BUFF ADDRESS
5277 022262 000          .BYTE    0        ;SECTOR 0
5278 022263 000          .BYTE    0        ;TRACK 0
5279 022264 000312          312           ;CYLINDER 312
5280
5281 022266 104417          TLOADRK        ;LOAD RK REGS
5282 022270 104424          TWAT32        ;WAIT FOR INTERRUPT
5283 022272 104002          ERROR    2      ;TO SLOW/NOT COMPLETE ERROR
5284
5285 022274 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5286 022276 104004          ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5287
5288          .SBTTL  **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT
5289

```

5290  
5291  
5292  
5293  
5294  
5295  
5296  
5297  
5298  
5299  
5300  
5301  
5302  
5303  
5304  
5305  
5306  
5307  
5308  
5309  
5310  
5311  
5312  
5313  
5314  
5315  
5316  
5317  
5318  
5319  
5320  
5321  
5322  
5323  
5324  
5325  
5326  
5327  
5328  
5329  
5330  
5331  
5332  
5333  
5334  
5335  
5336  
5337  
5338  
5339  
5340  
5341  
5342  
5343  
5344  
5345

```

*****
*TEST 72      MAXIMUM DATA TRANSFER (PART 1)
*
*   IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF
*   THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO
*   INSURE THE FORMAT IS CORRECT.
*
*   ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH
*   ONE SECTOR WRITES.  ISSUE A SEEK TO CYLINDER 0, TRACK 0.
*   ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000 WORDS
*   TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE CONTROLLER
*   TIME OUT IS NOT SET.  CHECK CYLINDER ADDRESS,
*   DISK ADDRESS, BUS ADDRESS AND WORD COUNT.  READ
*   EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.
*
*   NOTE:  THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
*   IN THE FIRST 256 SECTORS ON THE PACK.
*****

```

```

*****
TST72:  SCOPE
MOV      #5, $TIMES      ;; DO 5 ITERATIONS
BIT      #FPFMT, OPTFLG  ;; TEST IF FIRST PASS SWITCH SET
BNE      24$             ;; YES - SKIP FORMAT

CLRB     21$             ;; CLEAR ADDRESS POINTERS
CLR      22$

20$:    JSR      R4, LRLoad  ;; LOAD "L" REGISTERS
        WRHEAD  ;; WRITE HEADER
        -102    ;; 102 WORDS
        OBUF    ;; OBUF IS BUFF ADDRESS
        .BYTE   0        ;; SECTOR 0
21$:    .BYTE   0        ;; TRACK ADDRESS (VARIABLE)
22$:    0              ;; CYLINDER 0 (VARIABLE)

JSR      R4, GENCOM      ;; GO GENERATE HEADERS
1200     ;; WITH BAD SECTOR ERRORS

TLOADRK  ;; LOAD RK REGS
TWTAT112 ;; WAIT FOR INTERRUPT
ERROR    2          ;; TO SLOW/NOT COMPLETE ERROR

TCHKOP   ;; CHECK OPERATION FOR ANY ERRORS
ERROR    4 ;OR 5, 6, 7 ;; REPORT ALL ERRORS

CMPB     21$, #2        ;; TEST IF LAST TRACK
BEQ      23$           ;; YES - SKIP
INCB     21$           ;; ELSE BUMP TRACK
BR       20$           ;; LOOP

23$:    CLRB     21$     ;; CLEAR TRACK POINTER
CMP      22$, #3       ;; TEST IF LAST CYLINDER WRITTEN
BEQ      24$           ;; YES - SKIP
INC      22$           ;; ELSE BUMP CYLINDER
BR       20$           ;; LOOP

```

```

022300 000004
022302 012737 000005 001262
022310 032737 000400 001656
022316 001043
022320 105037 022343
022324 005037 022344
022330 004437 033324
022334 000127
022336 177676
022340 060446
022342 000
022343 000
022344 000000
022346 004437 037526
022352 001200
022354 104417
022356 104431
022360 104002
022362 104421
022364 104004
022366 123727 022343 000002
022374 001403
022376 105237 022343
022402 000752
022404 105037 022343
022410 023727 022344 000003
022416 001403
022420 005237 022344
022424 000741

```

## E11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(732) 01-OCT-76 10:50 PAGE 101  
 DZR6KB.P11 T72 MAXIMUM DATA TRANSFER (PART 1)

SEG 0134

```

5346 022426 013737 022440 001110 24$: MOV 1$, $LPERR ;SET LOCAL LOOP ON ERROR
5347 022434 012703 000400 MOV #400, R3 ;SET COUNT FOR SECTOR CLEARING
5348 022440 1$:
5349 022440 104416 TSSINIT ;CLEAR SUBSYSTEM
5350 022442 104003 ERROR 3 ;BAD INIT ERROR
5351 022444 004437 033324 JSR R4, LRLOAD ;LOAD "L" REGS
5352 022450 000123 WRDATA ;WRDATA
5353 022452 177400 -400 ;-400 WORDS
5354 022454 060446 OBUFF ;OBUFF IS BUFF ADDRESS
5355 022456 000 .BYTE 0 ;SECTOR 0
5356 022457 000 .BYTE 0 ;TRACK 0
5357 022460 000000 0 ;CYLINDER 0
5358
5359 022462 104417 TLOADRK ;LOAD RK REGS
5360 022464 104434 TWAT159 ;WAIT FOR INTERRUPT
5361 022466 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5362
5363 022470 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5364 022472 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5365
5366 022474 104415 SCOP1 ;LOCAL LOOP ON ERROR TO 1$
5367 022476 005303 DEC R3 ;DECREMENT COUNT
5368 022500 012762 060446 000004 2$: MOV #OBUFF, RKBA(R2) ;SET BA
5369 022506 012762 177400 000002 MOV #-400, RKWC(R2) ;AND WC AGAIN
5370 022514 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
5371 022520 013762 001626 000010 MOV DRVNUM, RKCS2(R2) ;SET DRIVE NUMBER
5372 022526 012762 000123 000000 MOV #WRDATA, RKCS1(R2) ;DO WRITE DATA
5373
5374 022534 104425 TWAT48 ;WAIT FOR INTERRUPT
5375 022536 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5376
5377 022540 032762 000200 000014 BIT #BSE, RKER(R2) ;BAD SECTOR ERROR?
5378 022546 001415 BEQ 3$ ;NO-SKIP
5379 022550 032737 000200 001656 BIT #BSE, OPTFLG ;TEST IF BSE TO MANY HAS BEEN REPORTED
5380 022556 001007 BNE 5$ ;YES - SKIP
5381 022560 052737 000200 001656 BIS #BSE, OPTFLG ;SET FLAG
5382 022566 012737 047423 001360 MOV #OPRO17, EMIN ;SET MESSAGE
5383 022574 104001 ERROR 1 ;"FIRST 256 SECTOR NOT BSE FREE"
5384 022576 000137 023132 5$: JMP 14$ ;GO TO EXIT
5385 3$: CHECK FOR ANY OTHER ERRORS
5386
5387 022602 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5388 022604 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5389 022606 104415 SCOP1 ;LOCAL LOOP TO 1$ (RESTART SECTOR CLEAR)
5390
5391 022610 005303 DEC R3 ;DECREMENT COUNT
5392 022612 001332 BNE 2$ ;LOOP IF NOT ZERO
5393
5394 022614 004437 033324 JSR R4, LRLOAD ;LOAD "L" REGS
5395 022620 000117 SEEK ;SEEK
5396 022622 000000 0 ;0 WORDS
5397 022624 000000 0 ;0 IS BUFF ADDRESS
5398 022626 000 .BYTE 0 ;SECTOR 0
5399 022627 000 .BYTE 0 ;TRACK 0
5400 022630 000000 0 ;CYLINDER 0
5401

```



5402	022632	104417		TLOADRK		;LOAD RK REGS
5403	022634	104423		TWAT16		;WAIT FOR INTERRUPT
5404	022636	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5405	022640	005037	001662	CLR	INTSET	;CLEAR FIRST INTERRUPT
5406	022644	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5407	022646	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
5408	022650	104427		TWAT80		;WAIT FOR SECOND INTERRUPT
5409	022652	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5410	022654	004437	037526	JSR	R4,GENCOM	;GENERATE DATA
5411	022660	004006		4006		;PATTERN 6, 1ST WORD REPEATED
5412	022662	000400		400		;400 WORDS
5413						
5414	022664		4\$:			
5415	022664	104416		TSSINIT		;CLEAR SUBSYSTEM
5416	022666	104003		ERROR	3	;BAD INIT ERROR
5417	022670	004437	033324	JSR	R4,LRLOAD	;LOAD "L" REGS
5418	022674	000123		WRDATA		;WRDATA
5419	022676	000000		0		;0 WORDS
5420	022700	060446		OBUFF		;OBUFF IS BUFF ADDRESS
5421	022702	000		.BYTE	0	;SECTOR 0
5422	022703	000		.BYTE	0	;TRACK 0
5423	022704	000000		0		;CYLINDER 0
5424						
5425	022706	052737	000020 001610	BIS	#BAI,L.CS2	
5426						
5427	022714	104417		TLOADRK		;LOAD RK REGS
5428	022716	104437		TWAT8S		;WAIT FOR SECOND INTERRUPT
5429	022720	104002		ERROR	2	;ELSE REPORT TO SLOW/NOT COMPLETE ERROR
5430						
5431	022722		7\$:			
5432	022722	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5433	022724	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5434						
5435	022726	104415		SCOP1		;INTERNAL LOOP ON ERROR TO 4\$
5436	022730	012703	000400	MOV	#400,R3	;SET COUNTER FOR READ-COMPARE LOOP
5437	022734	005037	022776	CLR	10\$	;CLEAR SECTOR AND TRACK VALUES
5438	022740	005037	023000	CLR	12\$	;CLEAR CYL VALUE
5439	022744	013737	022760 001110	MOV	8\$,SLPERR	;SET LOCAL LOOP ON ERROR
5440	022752	042737	000020 001610	BIC	#BAI,L.CS2	;CLEAR BAI
5441						
5442	022760		8\$:			
5443	022760	104416		TSSINIT		;CLEAR SUBSYSTEM
5444	022762	104003		ERROR	3	;BAD INIT ERROR
5445	022764	004437	033324	JSR	R4,LRLOAD	;LOAD RK REGS
5446	022770	000121		RDDATA		;READ DATA
5447	022772	177400		-400		;400 WORDS
5448	022774	056446		IBUFF		;INTO IBUFF
5449	022776	000	10\$:	.BYTE	0	;SECTOR (VARIABLE)
5450	022777	000	11\$:	.BYTE	0	;TRACK (VARIABLE)
5451	023000	000000	12\$:	0		;CYL (VARIABLE)
5452						
5453	023002	104417		TLOADRK		;LOAD RK REGS
5454	023004	104425		TWAT48		;WAIT FOR INTERRUPT
5455	023006	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5456						
5457	023010	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS

```

5458 023012 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5459
5460 023014 104415          SCOPI          ;LOCAL LOOP ON ERROR TO 8$
5461
5462 023016 004437 037526    JSR          R4,GENCOM      ;COMPARE DATA
5463 023022 100000          100000
5464 023024 000400          400          ;400 WORDS
5465 023026 000413          BR          13$          ;NO MISCOMPARE-EXIT LOOP
5466 023030 104015          ERROR 15          ;REPORT FIRST ERROR
5467
5468 023032 013700 001634    MOV          EARLMT,R0      ;GET ERROR LIMIT
5469 023036 005300          64$: DEC        R0          ;DECREMENT COUNT
5470 023040 001406          BEQ        65$          ;IF ZERO - EXIT
5471 023042 004437 037526    JSR          R4,GENCOM      ;CONTINUE DATA COMPARE
5472 023046 040000          40000
5473 023050 000402          BR          65$          ;NO MORE ERRORS - EXIT
5474 023052 104016          ERROR 16          ;REPORT NEXT ERROR
5475 023054 000770          BR          64$          ;LOOP
5476 023056          65$:
5477
5478 023056 104415          13$: SCOPI          ;LOCAL LOOP TO 8$
5479
5480 023060 005303          DEC        R3          ;DEC READ LOOP COUNT
5481 023062 001423          BEQ        14$          ;IF ZERO-EXIT
5482
5483 023064 105237 022776    INCB       10$          ;BUMP SECTOR
5484 023070 123727 022776 000026  CMPB      10$,#26       ;FINISHED WITH TRACK?
5485 023076 001332          BNE        9$          ;NO-LOOP
5486 023100 105037 022776    CLRB      10$          ;CLEAR SECTOR
5487 023104 105237 022777    INCB       11$          ;BUMP TRACK
5488 023110 123727 022777 000003  CMPB      11$,#3       ;FINISHED WITH CYLINDER?
5489 023116 001322          BNE        9$          ;NO-LOOP
5490 023120 105037 022777    CLRB      11$          ;CLEAR TRACK
5491 023124 005237 023000    INC        12$          ;BUMP CYL.
5492 023130 000715          BR          9$          ;LOOP
5493
5494 023132          14$:
5495 *****
5496 ;TEST 73          MAXIMUM DATA TRANSFER (PART 2)
5497 ;
5498 ;          ZERO OUT FIRST 256 SECTORS OF THE DISK WITH
5499 ;          20000 WORD WRITE.  SEEK TO CYLINDER 632.
5500 ;          ISSUE A WRITE OF MAXIMUM DATA TRANSFER
5501 ;          20000 WORD WRITE.  MAKE SURE CONTROLLER TIME
5502 ;          OUT IS NOT SET.  CHECK CYLINDER ADDRESS
5503 ;          DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
5504 ;          SEEK TO CYLINDER 632.  ISSUE A WRITE CHECK
5505 ;          OF 20000 WORDS.  MAKE SURE NO ERROR SETS.
5506 ;
5507 ;          NOTE:  THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
5508 ;          IN THE FIRST 256 SECTORS ON THE PACK.
5509 ;
5510 *****
5511 023132 000004          TST73: SCOPE
5512 023134 012737 000005 001262  MOV        #5,$TIMES      ;DO 5 ITERATIONS
5513 023142 104416          TSSINIT          ;CLEAR SUBSYSTEM
    
```

# H11

5514	023144	104003			ERROR 3	;BAD INIT ERROR
5515	023146	012700	000620		MOV #400.,RO	;SET COUNT FOR INTERRUPT WAIT
5516	023152	005037	060446		CLR OBUFF	
5517						
5518	023156	004437	033324		JSR R4,LRLOAD	;LOAD "L" REGS
5519	023162	000123			WRDATA	;WRDATA
5520	023164	000000			0	;0 WORDS
5521	023166	060446			OBUFF	;OBUFF IS BUFF ADDRESS
5522	023170	000			.BYTE 0	;SECTOR 0
5523	023171	000			.BYTE 0	;TRACK 0
5524	023172	000000			0	;CYLINDER 0
5525	023174	052737	000020	001610	BIS #BAI,L.CS2	;SET BAI
5526	023202	104417			TLOADRK	;LOAD RK REGS
5527	023204	104434		1\$:	TWAT159	;WAIT FOR INTERRUPT
5528	023206	000401			BR 2\$	;NO INTERRUPT-SKIP
5529	023210	000403			BR 3\$	;INTERRUPT-SKIP
5530						
5531	023212	005300		2\$:	DEC RO	;DEC WAIT COUNTER
5532	023214	001373			BNE 1\$	;NO ZERO-LOOP
5533	023216	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
5534						
5535	023220	032762	000200	000014	3\$: BIT #BSE,RKER(R2)	;DID BSE SET
5536	023226	001415			BEG 4\$	;NO-SKIP
5537						
5538	023230	032737	000200	001656	BIT #BSERPT,OPTFLG	;TEST IF TO MANY BAD SECTORS REPORTED
5539	023236	001007			BNE 12\$	;YES - SKIP
5540	023240	052737	000200	001656	BIS #BSERPT,OPTFLG	;SET FLAG
5541	023246	012737	047423	001360	MOV #OPRO17,EMIN	;SET MESSAGE
5542	023254	104001			ERROR 1	; "FIRST 256 SECTORS NOT BSE FREE"
5543	023256	000137	023610	12\$:	JMP 11\$	;EXIT
5544						
5545	023262			4\$:		
5546	023262	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
5547	023264	104004			ERROR 4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
5548						
5549	023266	004437	033324		JSR R4,LRLOAD	;LOAD "L" REGS
5550	023272	000117			SEEK	;SEEK
5551	023274	000000			0	;0 WORDS
5552	023276	000000			0	;0 IS BUFF ADDRESS
5553	023300	000			.BYTE 0	;SECTOR 0
5554	023301	000			.BYTE 0	;TRACK 0
5555	023302	000632			632	;CYLINDER 632
5556						
5557	023304	104417			TLOADRK	;LOAD RK REGS
5558	023306	104423			TWAT16	;WAIT FOR INTERRUPT
5559	023310	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
5560	023312	005037	001662		CLR INTSET	;CLEAR INTERRUPT FLAG
5561						
5562	023316	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
5563	023320	104004			ERROR 4 ;OR 5, 6, 7	;REPORT ALL ERRORS
5564						
5565	023322	104427			TWAT80	;WAIT FOR 2ND INTERRUPT
5566	023324	104002			ERROR 2	
5567						
5568	023326	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
5569	023330	104004			ERROR 4 ;OR 5, 6, 7	;REPORT ALL ERRORS

```

5570
5571 023332 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5572 023336 000105 CLEAR ;CLEAR
5573 023340 000000 0 ;0 WORDS
5574 023342 000000 0 ;0 IS BUFF ADDRESS
5575 023344 000 .BYTE 0 ;SECTOR 0
5576 023345 000 .BYTE 0 ;TRACK 0
5577 023346 000000 0 ;CYLINDER 0
5578
5579 023350 104417 TLOADRK ;LOAD RK REGS
5580 023352 104423 TWAT16 ;WAIT FOR INTERRUPT
5581 023354 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5582
5583 023356 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5584 023360 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
5585
5586 023362 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5587 023366 000123 WRDATA ;WRDATA
5588 023370 000000 0 ;0 WORDS
5589 023372 060446 OBUFF ;OBUFF IS BUFF ADDRESS
5590 023374 000 .BYTE 0 ;SECTOR 0
5591 023375 000 .BYTE 0 ;TRACK 0
5592 023376 000000 0 ;CYLINDER 0
5593
5594 023400 012737 135143 060446 MOV #135143,OBUFF ;SET WORD FOR OUTPUT
5595 023406 012700 000621 MOV #401,R0 ;SET COUNT FOR INTERRUPT WAIT
5596 023412 052737 000020 001610 BIS #BAI,L.CS2 ;SET BUS ADDRESS INC INHIBIT
5597
5598 023420 104417 TLOADRK ;LOAD RK REGS
5599 023422 104434 TWAT159 ;WAIT FOR INTERRUPT
5600 023424 000401 BR 6$ ;NO INTERRUPT-BRANCH
5601 023426 000403 BR 7$ ;INTERRUPT-BRANCH
5602
5603 023430 005300 6$: DEC R0 ;DEC WAIT COUNT
5604 023432 001373 BNE 5$ ;LOOP IF NOT ZERO
5605 023434 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5606
5607 023436 7$:
5608 023436 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5609 023440 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5610 023442 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5611 023446 000117 SEEK ;SEEK
5612 023450 000000 0 ;0 WORDS
5613 023452 000000 0 ;0 IS BUFF ADDRESS
5614 023454 000 .BYTE 0 ;SECTOR 0
5615 023455 000 .BYTE 0 ;TRACK 0
5616 023456 000632 632 ;CYLINDER 632
5617 023460 104417 TLOADRK ;LOAD RK REGS
5618 023462 104423 TWAT16 ;WAIT FOR INTERRUPT
5619 023464 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5620 023466 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
5621
5622 023472 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5623 023474 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
5624
5625 023476 104427 TWAT80 ;WAIT FOR SECOND INIT

```

```

5626 023500 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
5627 023502 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5628 023504 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
5629
5630 023506 004437 033324    JSR    R4,LRLOAD ;LOAD "L" REGS
5631 023512 000105          CLEAR          ;CLEAR
5632 023514 000000          0              ;0 WORDS
5633 023516 000000          0              ;0 IS BUFF ADDRESS
5634 023520 000          .BYTE 0         ;SECTOR 0
5635 023521 000          .BYTE 0         ;TRACK 0
5636 023522 000000          0              ;CYLINDER 0
5637
5638 023524 104417          TLOADRK        ;LOAD RK REGS
5639 023526 104423          TWAT16        ;WAIT FOR INTERRUPT
5640 023530 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
5641
5642 023532 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5643 023534 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
5644
5645 023536 004437 033324    JSR    R4,LRLOAD ;LOAD "L" REGS
5646 023542 000131          WRTCHK        ;WRTCHK
5647 023544 000000          0              ;0 WORDS
5648 023546 060446          OBUFF         ;OBUFF IS BUFF ADDRESS
5649 023550 000          .BYTE 0         ;SECTOR 0
5650 023551 000          .BYTE 0         ;TRACK 0
5651 023552 000000          0              ;CYLINDER 0
5652 023554 052737 000020 001610  BIS    #BAI,L.CS2 ;SET BAI FLAG
5653 023562 012700 000621          MOV     #401.,R0 ;SET WAIT COUNT
5654
5655 023566 104417          TLOADRK        ;LOAD RK REGS
5656 023570 104434          TWAT159       ;WAIT FOR INTERRUPT
5657 023572 000401          BR          9$  ;NO INTERRUPT-SKIP
5658 023574 000403          BR          10$ ;INTERRUPT-SKIP
5659
5660 023576 005300          9$: DEC     R0    ;DEC WAIT COUNT
5661 023600 001373          BNE     8$    ;NOT ZERO-LOOP
5662 023602 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
5663
5664 023604          10$:
5665 023604 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5666 023606 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5667
5668 023610          11$:
5669          ;*****
5670          ;*TEST 74          CONTROLLER TIME OUT
5671          ;*
5672          ;*          SEEK TO CYLINDER 632.  ISSUE A RECALIBRATE AND DO NOT
5673          ;*          WAIT FOR SECOND INTERRUPT.  NOW ISSUE A READ HEADER
5674          ;*          OF CYLINDER 0, TRACK 0.  MAKE SURE CONTROLLER TIME
5675          ;*          OUT SETS.
5676          ;*
5677          ;*****
5678 023610 000004          TST74: SCOPE
5679 023612 012737 000005 001262  MOV     #5.,$TIMES ;DO 5. ITERATIONS
5680 023620 104416          TSSINIT       ;CLEAR SUBSYSTEM
5681 023622 104003          ERROR 3          ;BAD INIT ERROR

```



5682								
5683	023624	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
5684	023630	000117		SEEK				;SEEK
5685	023632	000000		0				;0 WORDS
5686	023634	000000		0				;0 IS BUFF ADDRESS
5687	023636	000		.BYTE	0			;SECTOR 0
5688	023637	000		.BYTE	0			;TRACK 0
5689	023640	000632		632				;CYLINDER 632
5690								
5691	023642	104417		TLOADRK				;LOAD RK REGS
5692	023644	104423		TWAT16				;WAIT FOR INTERRUPT
5693	023646	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
5694								
5695	023650	104421		TCHKOP				;CHECK OPERATION FOR ANY ERRORS
5696	023652	104004		ERROR	4 ;OR 5, 6, 7			;REPORT ALL ERRORS
5697								
5698	023654	005037	001662	CLR	INTSET			;CLEAR INTERRUPT FLAG
5699	023660	104427		TWAT80				;WAIT FOR SECOND INTERRUPT
5700	023662	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
5701	023664	104421		TCHKOP				;CHECK OPERATION FOR ANY ERRORS
5702	023666	104004		ERROR	4 ;OR 5, 6, 7			;REPORT ALL ERRORS
5703								
5704	023670	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
5705	023674	000105		CLEAR				;CLEAR
5706	023676	000000		0				;0 WORDS
5707	023700	000000		0				;0 IS BUFF ADDRESS
5708	023702	000		.BYTE	0			;SECTOR 0
5709	023703	000		.BYTE	0			;TRACK 0
5710	023704	000000		0				;CYLINDER 0
5711								
5712	023706	104417		TLOADRK				;LOAD RK REGS
5713	023710	104423		TWAT16				;WAIT FOR INTERRUPT
5714	023712	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
5715								
5716	023714	104421		TCHKOP				;CHECK OPERATION FOR ANY ERRORS
5717	023716	104004		ERROR	4 ;OR 5, 6, 7			;REPORT ALL ERRORS
5718								
5719	023720	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
5720	023724	000113		RECAL				;RECAL
5721	023726	000000		0				;0 WORDS
5722	023730	000000		0				;0 IS BUFF ADDRESS
5723	023732	000		.BYTE	0			;SECTOR 0
5724	023733	000		.BYTE	0			;TRACK 0
5725	023734	000000		0				;CYLINDER 0
5726								
5727	023736	104417		TLOADRK				;LOAD RK REGS
5728	023740	104423		TWAT16				;WAIT FOR INTERRUPT
5729	023742	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
5730								
5731	023744	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
5732	023750	000125		RDHEAD				;RDHEAD
5733	023752	000000		0				;0 WORDS
5734	023754	000000		0				;0 IS BUFF ADDRESS
5735	023756	000		.BYTE	0			;SECTOR 0
5736	023757	000		.BYTE	0			;TRACK 0
5737	023760	000000		0				;CYLINDER 0

```

5738
5739 023762 104417 TLOADRK ;LOAD RK REGS
5740 023764 104436 TWAT25 ;WAIT FOR INTERRUPT
5741 023766 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5742
5743 023770 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR
5744 023772 000000 0
5745 023774 000000 0
5746 023776 000002 CTOERR ;CONTROLLER TIME OUT
5747 024000 104004 ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
5748 024002 104416 TSSINIT ;CLEAR SUBSYSTEM
5749 024004 104003 ERROR 3 ;BAD INIT ERROR
5750 024006 005037 001662 CLR INTSET ;CLEAR INT FLAG
5751 024012 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INT ENABLE
5752 024020 104437 TWAT8S ;WAIT FOR SECOND INT
5753 024022 104002 ERROR 2

```

.SBTTL \*\*ERRORS DURING DATA TRANSFER

```

*****
*TEST 75 LIMIT DETECT ON DATA TRANSFER
*****
*
* ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
* A SEEK TO CYLINDER 2 WITH BAD PARITY. ISSUE A DRIVE
* CLEAR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
* TRACK 0, HEAD 0. SEEK INCOMPLETE BECAUSE OF OUTER
* LIMIT SHOULD BE THE ONLY ERROR SET.
*****

```

```

5769 024024 000004 TST75: SCOPE
5770 024026 012737 000012 001262 MOV #10.,$TIMES ;;DO 10. ITERATIONS
5771
5772 024034 104416 TSSINIT ;CLEAR SUBSYSTEM
5773 024036 104003 ERROR 3 ;BAD INIT ERROR
5774
5775 024040 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5776 024044 000113 RECAL ;RECAL
5777 024046 000000 0 ;0 WORDS
5778 024050 000000 0 ;0 IS BUFF ADDRESS
5779 024052 000 .BYTE 0 ;SECTOR 0
5780 024053 000 .BYTE 0 ;TRACK 0
5781 024054 000000 0 ;CYLINDER 0
5782
5783 024056 104417 TLOADRK ;LOAD RK REGS
5784 024060 104423 TWAT16 ;WAIT FOR INTERRUPT
5785 024062 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5786
5787 024064 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
5788 024070 104437 TWAT8S ;WAIT FOR SECOND INTERRUPT
5789 024072 104002 ERROR 2
5790
5791 024074 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5792 024076 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
5793

```

```

5794 024100 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5795 024104 000117 SEEK ;SEEK
5796 024106 000000 0 ;0 WORDS
5797 024110 000000 0 ;0 IS BUFF ADDRESS
5798 024112 000 .BYTE 0 ;SECTOR 0
5799 024113 000 .BYTE 0 ;TRACK 0
5800 024114 000002 2 ;CYLINDER 2
5801 024116 012737 000020 001616 MOV #PAT,L.MR1 ;SET EVEN PARITY BIT
5802 024124 104417 TLOADRK ;LOAD RK REGS
5803 024126 104423 TWAT16 ;WAIT FOR INTERRUPT
5804 024130 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5805
5806 024132 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
5807
5808 024140 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5809 024144 000105 CLEAR ;CLEAR
5810 024146 000000 0 ;0 WORDS
5811 024150 000000 0 ;0 IS BUFF ADDRESS
5812 024152 000 .BYTE 0 ;SECTOR 0
5813 024153 000 .BYTE 0 ;TRACK 0
5814 024154 000000 0 ;CYLINDER 0
5815
5816 024156 005037 001616 CLR L.MR1 ;CLEAR PARITY BIT
5817
5818 024162 104417 TLOADRK ;LOAD RK REGS
5819 024164 104423 TWAT16 ;WAIT FOR INTERRUPT
5820 024166 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5821
5822 024170 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
5823 024174 000123 WRDATA ;WRDATA
5824 024176 177400 -400 ;-400 WORDS
5825 024200 060446 OBUFF ;OBUFF IS BUFF ADDRESS
5826 024202 000 .BYTE 0 ;SECTOR 0
5827 024203 000 .BYTE 0 ;TRACK 0
5828 024204 000001 1 ;CYLINDER 1
5829
5830 024206 104417 TLOADRK ;LOAD RK REGS
5831 024210 104423 TWAT16 ;WAIT FOR INTERRUPT
5832 024212 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5833
5834 024214 104422 TCHKWE ;CHECK OPERATION WITH ERROR
5835 024216 000002 SKIERR ;SEEK INCOMPLETE
5836 024220 000000 0
5837 024222 000000 0
5838 024224 104004 ERROR 4 ;OR 5,6,OR7 ;REPORT ALL DISCREPANCIES
5839
5840 ; THIS DELAY IS NEEDED TO ALLOW SEEK AND NO MOTION TO
5841 ; SET IN THE DRIVE BEFORE THE SUBSYSTEM CLEAR IS DONE.
5842
5843 024226 012701 025000 MOV #25000,R1 ;SET A COUNT
5844 024232 005301 DEC R1 ;DEC COUNT
5845 024234 001376 BNE 3$ ;LOOP UNTIL ZERO
5846
5847 024236 104416 TSSINIT ;CLEAR SUBSYSTEM
5848 024240 104003 ERROR 3 ;BAD INIT ERROR
5849

```



```

5850 024242 012762 000100 000000      MOV      #IE,RKCS1(R2)  ;SET IE
5851
5852 024250 005037 001662      CLR      INTSET        ;CLEAR INT FLAG
5853 024254 104437      TWATBS      ;WAIT FOR SECOND INTERRUPT
5854 024256 000401      BR       1$
5855 024260 000404      BR       2$
5856
5857 024262 012737 055036 001372 1$:      MOV      #DH016,DH2N   ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
5858                                ;ALLOWING HEADS TO RELOAD"
5859 024270 104002      ERROR    2
5860
5861 024272      2$:
5862
5863      ;*****
5864      ;*TEST 76      PROGRAMMING ERROR
5865      ;*
5866      ;*      ISSUE A SUBSYSTEM CLEAR. ISSUE
5867      ;*      A READ DATA OF 400 WORDS ON CYLINDER 0,
5868      ;*      TRACK 0, SECTOR 0. DURING READ ISSUE A
5869      ;*      WRITE TO THE SPARE REGISTER. MAKE SURE
5870      ;*      PROGRAMMING ERROR SETS.
5871      ;*****
5871 024272 000004      TST76:  SCOPE
5872 024274 012737 000062 001262      MOV      #50.,$TIMES  ;;DO 50. ITERATIONS
5873 024302 104416      TSSINIT      ;CLEAR SUBSYSTEM
5874 024304 104003      ERROR    3          ;BAD INIT ERROR
5875
5876 024306 004437 033324      JSR      R4,LRLOAD   ;LOAD "L" REGS
5877 024312 000121      RDDATA      ;RDDATA
5878 024314 177400      -400        ;-400 WORDS
5879 024316 056446      Ibuff      ;IBUFF IS BUFF ADDRESS
5880 024320 000        .BYTE    0          ;SECTOR 0
5881 024321 000        .BYTE    0          ;TRACK 0
5882 024322 000000      0          ;CYLINDER 0
5883
5884 024324 104417      TLOADRK      ;LOAD RK REGS
5885
5886 024326 012762 000001 000022      MOV      #1,RKSPAR(R2) ;WRITE SPARE REGISTER
5887
5888 024334 104423      TWAT16      ;WAIT FOR INTERRUPT
5889 024336 104002      ERROR    2          ;TO SLOW/NOT COMPLETE ERROR
5890
5891 024340 104422      TCHKWE      ;CHECK OPERATION WITH EXPECTED ERROR
5892 024342 000000      0
5893 024344 000000      0
5894 024346 000020      PGERR
5895 024350 104004      ERROR    4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
5896
5897
5898      ;*****
5899      ;*TEST 77      ECC HARD
5900      ;*
5901      ;*      ISSUE A SUBSYSTEM CLEAR. ISSUE
5902      ;*      A WRITE DATA WORDS CONSISTING OF 177777 TO
5903      ;*      CYLINDER 0, TRACK 0, SECTOR 0. NOW WRITE
5904      ;*      ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0.
5905      ;*      DURING WRITE ISSUE CONTROLLER CLEAR. MAKE
5906      ;*      SURE PROGRAMMING ERROR IS RESET. NOW
  
```

```

5906          : *      ISSUE A READ DATA TO CYLINDER 0, TRACK 0,
5907          : *      HEAD 0 AND AN ECC HARD ERROR SHOULD SET.
5908          : *****
5909 024352 000004          TST77: SCOPE
5910 024354 012737 000062 001262      MOV      #50.,$TIMES      ;DO 50. ITERATIONS
5911 024362 104416          TSSINIT          ;CLEAR SUBSYSTEM
5912 024364 104003          ERROR      3          ;BAD INIT ERROR
5913
5914 024366 004437 037526          JSR      R4,GENCOM      ;GENERATE DATA OF ALL ONES
5915 024372 000007          7
5916 024374 000400          400
5917
5918 024376 004437 033324          JSR      R4,LRLOAD      ;LOAD "L" REGS
5919 024402 000123          WRDATA          ;WRDATA
5920 024404 177400          -400          ;-400 WORDS
5921 024406 060446          OBUF          ;OBUF IS BUFF ADDRESS
5922 024410          0          ;SECTOR 0
5923 024411          0          ;TRACK 0
5924 024412 000000          0          ;CYLINDER 0
5925
5926 024414 104417          TLOADRK          ;LOAD RK REGS
5927
5928 024416 005737 001662          1$:  TST      INTSET          ;CHECK IF INTERRUPT HAS OCCURRED
5929 024422 001027          BNE      2$          ;YES - MUCH TO SOON. REPORT ERROR
5930 024424 026227 000002 177630      CMP      RKWC(R2), #-150 ;CHECK THAT MORE THAN ONE SILO FULL
5931 024432 103771          BLO      1$          ;OF DATA HAS BEEN WRITTEN:ELSE LOOP
5932
5933 024434 052762 100000 000000      BIS      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER (CROWBAR WRITE)
5934
5935 024442 004437 033324          JSR      R4,LRLOAD      ;LOAD "L" REGS
5936 024446 000121          RDATA          ;RDATA
5937 024450 177400          -400          ;-400 WORDS
5938 024452 056446          IBUF          ;IBUF IS BUFF ADDRESS
5939 024454          0          ;SECTOR 0
5940 024455          0          ;TRACK 0
5941 024456 000000          0          ;CYLINDER 0
5942
5943 024460 104417          TLOADRK          ;LOAD RK REGS
5944 024462 104425          TWAT48          ;WAIT FOR INTERRUPT
5945 024464 104002          ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
5946
5947 024466 104422          TCHKWE          ;CHECK OPERATION WITH ERROR
5948 024470 000000          0
5949 024472 000003          DCKERR!ECHERR      ;DATA CHECK AND ECC HARD
5950 024474 000000          0
5951 024476 104004          ERROR      4 ;OR 5,6,7      ;REPORT ALL DISCREPANCIES
5952
5953 024500 000402          BR      3$          ;SKIP TO EXIT
5954
5955 024502          2$:  TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5956 024504 104421          ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5957 024506          3$:
5958
5959
5960          : *****
5961          : *TEST 100      DRIVE TIMING ERROR

```

```

5962      : *      ISSUE A SUBSYSTEM CLEAR. SEEK TO CYLINDER 632.
5963      : *      ISSUE A RECALIBRATE BUT DO NOT WAIT FOR SECOND INTERRUPT.
5964      : *      PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER
5965      : *      OF CYLINDER 0, TRACK 0. CLOCK THROUGH SEEK
5966      : *      AND DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE.
5967      : *      DRIVE TIMING ERROR SHOULD SET BECAUSE OF NO DATA
5968      : *      TRANSITIONS ON DATA LINE.
5969      : *      *****
5970 024506 000004      †ST100: SCOPE
5971 024510 012737 000012 001262      MOV      #10.,$TIMES      ;;DO 10. ITERATIONS
5972
5973 024516 104416      TSSINIT      ;CLEAR SUBSYSTEM
5974 024520 104003      ERROR 3      ;BAD INIT ERROR
5975
5976 024522 004437 033324      JSR      R4,LRLOAD      ;LOAD "L" REGS
5977 024526 000117      SEEK      ;SEEK
5978 024530 000000      0      ;0 WORDS
5979 024532 000000      0      ;0 IS BUFF ADDRESS
5980 024534      000      .BYTE 0      ;SECTOR 0
5981 024535      000      .BYTE 0      ;TRACK 0
5982 024536 000632      632      ;CYLINDER 632
5983
5984 024540 104417      TLOADRK      ;LOAD RK REGS
5985 024542 104423      TWAT16      ;WAIT FOR INTERRUPT
5986 024544 104002      ERROR 2      ;TC SLOW/NOT COMPLETE ERROR
5987 024546 005037 001662      CLR      INTSET      ;CLEAR INT FLAG
5988 024552 104430      TWAT96      ;WAIT FOR SECOND INTERRUPT
5989 024554 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
5990
5991 024556 004437 033324      JSR      R4,LRLOAD      ;LOAD "L" REGS
5992 024562 000113      RECAL      ;RECAL
5993 024564 000000      0      ;0 WORDS
5994 024566 000000      0      ;0 IS BUFF ADDRESS
5995 024570      000      .BYTE 0      ;SECTOR 0
5996 024571      000      .BYTE 0      ;TRACK 0
5997 024572 000000      0      ;CYLINDER 0
5998
5999 024574 104417      TLOADRK      ;LOAD RK REGS
6000 024576 104423      TWAT16      ;WAIT FOR INTERRUPT
6001 024600 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
6002
6003 024602 004437 033324      JSR      R4,LRLOAD      ;LOAD "L" REGS
6004 024606 000125      RDHEAD      ;RDHEAD
6005 024610 000000      0      ;0 WORDS
6006 024612 000000      0      ;0 IS BUFF ADDRESS
6007 024614      000      .BYTE 0      ;SECTOR 0
6008 024615      000      .BYTE 0      ;TRACK 0
6009 024616 000000      0      ;CYLINDER 0
6010 024620 012737 000040 001616      MOV      #DMD,L.MR1      ;SET DIAG MODE
6011 024626 104417      TLOADRK      ;LOAD RK REGS
6012
6013 024630 004437 033600      JSR      R4,MCLOCK      ;CLOCK CONTROLLER THROUGH SEEK
6014 024634 001062      1062      ;AND CLEAR TO READ
6015
6016 024636 005062 000026      CLR      RKMRI(R2)      ;RESET DIAG MODE
6017

```

```

6018 024642 104424          TWAT32          ;WAIT FOR INTERRUPT
6019 024644 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6020 024646 104422          TCHKWE          ;CHECK OPERATION WITH EXP ERROR
6021 024650 010000          DTERR          ;DRIVE TIMING ERROR
6022 024652 000000          0
6023 024654 000000          0
6024 024656 104004          ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
6025
6026 024660          IS:
6027 024660 104416          TSSINIT          ;CLEAR SUBSYSTEM
6028 024662 104003          ERROR 3          ;BAD INIT ERROR
6029 024664 012762 000100 000000  MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
6030 024672 005037 001662          CLR INTSET       ;CLEAR INT FLAG
6031
6032 024676 104437          TWATBS          ;WAIT FOR INTERRUPT FOR END OF RECAL
6033 024700 104002          ERROR 2
6034
6035 024702 004437 033324          JSR R4,LRLOAD   ;LOAD "L" REGS
6036 024706 000105          CLEAR          ;CLEAR
6037 024710 000000          0             ;0 WORDS
6038 024712 000000          0             ;0 IS BUFF ADDRESS
6039 024714 000          .BYTE 0         ;SECTOR 0
6040 024715 000          .BYTE 0         ;TRACK 0
6041 024716 000000          0             ;CYLINDER 0
6042
6043 024720 104417          TLOADRK          ;LOAD RK REGS
6044 024722 104423          TWAT16          ;WAIT FOR INTERRUPT
6045 024724 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6046 024726 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6047 024730 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS

```

.SBTTL \*\*ERROR FORCING IN DRIVE

```

*****
*TEST 101 INITIALIZE CLEARING SACK
*
* ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
* DRIVE. ISSUE A SUBSYSTEM CLEAR. PUT CONTROLLER IN
* DIAGNOSTIC MODE. ISSUE A SELECT COMMAND WITH
* MESSAGE ID = 3 AND DRIVE SELECTED = 0. CLOCK THROUGH
* PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE. MAKE
* SURE UNIT FIELD ERROR DOES NOT SET.
*****

```

```

6061
6062 024732 000004          ST101: SCOPE
6063 024734 012737 000062 001262  MOV #50.,STIMES ;DO 50. ITERATIONS
6064 024742 104416          TSSINIT          ;CLEAR SUBSYSTEM
6065 024744 104003          ERROR 3          ;BAD INIT ERROR
6066
6067 024746 004437 033324          JSR R4,LRLOAD   ;LOAD "L" REGS
6068 024752 000101          SELDRV          ;SELDV
6069 024754 000000          0             ;0 WORDS
6070 024756 000000          0             ;0 IS BUFF ADDRESS
6071 024760 000          .BYTE 0         ;SECTOR 0
6072 024761 000          .BYTE 0         ;TRACK 0
6073 024762 000000          0             ;CYLINDER 0

```

```

6074
6075 024764 104417 TLOADRK ;LOAD RK REGS
6076 024766 104423 TWAT16 ;WAIT FOR INTERRUPT
6077 024770 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6078
6079 024772 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6080 024774 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6081
6082 024776 104416 TSSINIT ;CLEAR SUBSYSTEM
6083 025000 104003 ERROR 3 ;BAD INIT ERROR
6084
6085 025002 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6086 025006 000101 SELDRV ;SELDV
6087 025010 000000 0 ;0 WORDS
6088 025012 000000 0 ;0 IS BUFF ADDRESS
6089 025014 000 .BYTE 0 ;SECTOR 0
6090 025015 000 .BYTE 0 ;TRACK 0
6091 025016 000000 0 ;CYLINDER 0
6092 025020 012737 000043 001616 MOV #3!DMD,L.MR1 ;SET DIAG MODE AND MESSAGE PAIR 3
6093 025026 005037 001610 CLR L.CS2 ;SELECT DRIVE 0
6094
6095 025032 104417 TLOADRK ;LOAD RK REGS
6096
6097 025034 004437 033600 JSR R4,MCLOCK ;CLOCK THROUGH PHASE ADDRESS 6
6098 025040 001027 1027
6099
6100 025042 042762 000040 000026 BIC #DMD,RKMR1(R2) ;CLEAR MAINTENANCE MODE
6101
6102 025050 104424 TWAT32 ;WAIT FOR INTERRUPT
6103 025052 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
6104
6105 025054 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6106 025056 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6107
6108 ;*****
6109 ;*TEST 102 DRIVE OFF TRACK
6110 ;*
6111 ;* ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
6112 ;* OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC
6113 ;* MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
6114 ;* TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR
6115 ;* MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK
6116 ;* SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILIABLE DRIVES.
6117 ;*
6118 ;*****
6118 025060 000004 ST102: SCOPE
6119 025062 012737 000012 001262 MOV #10.,STIMES ;DO 10. ITERATIONS
6120 025070 104416 TSSINIT ;CLEAR SUBSYSTEM
6121 025072 104003 ERROR 3 ;BAD INIT ERROR
6122
6123 025074 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6124 025100 000113 RECAL ;RECAL
6125 025102 000000 0 ;0 WORDS
6126 025104 000000 0 ;0 IS BUFF ADDRESS
6127 025106 000 .BYTE 0 ;SECTOR 0
6128 025107 000 .BYTE 0 ;TRACK 0
6129 025110 000000 0 ;CYLINDER 0

```



```

6186 025250 104417 TLOADRK ;LOAD RK REGS
6187 025252 104423 TWAT16 ;WAIT FOR INTERRUPT
6188 025254 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6189
6190 025256 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6191 025260 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6192
6193 025262 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6194 025266 000123 WRDATA ;WRDATA
6195 025270 177777 -1 ;-1 WORDS
6196 025272 060446 OBUFF ;OBUFF IS BUFF ADDRESS
6197 025274 000 .BYTE 0 ;SECTOR 0
6198 025275 000 .BYTE 0 ;TRACK 0
6199 025276 000000 0 ;CYLINDER 0
6200 025300 012737 000040 001616 MOV #DMD,L.MR1 ;SET DIAGNOSTIC MODE
6201
6202 025306 104417 TLOADRK
6203
6204 025310 004437 033600 JSR R4,MCLOCK ;CLOCK THROUGH SEEK & DRIVE CLEAR
6205 025314 001064 1064
6206
6207 025316 005062 000026 CLR RKMR1(R2) ;CLEAR DIAGNOSTIC MODE
6208 025322 104424 TWAT32 ;WAIT FOR INTERRUPT
6209 025324 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
6210
6211 ;*****
6212 ;*****
6213
6214 ; NOTE:
6215 ;
6216 ; THE DRIVE LOGIC DOES NOT RAISE ATTENTION AT THE OCCURRENCE
6217 ; OF THE ERROR. IT FIRST RETURNS THE HEADS TO CENTERLINE
6218 ; AND FIRES A HEADS SETTLING ONE-SHOT. ATTENTION IS NOT SET
6219 ; UNTIL THE ONE-SHOT TIMES OUT. CONSEQUENTLY THE CONTROLLER
6220 ; WILL FINISH THE WRITE, RAISE INTERRUPT AS THOUGH NO ERROR
6221 ; OCCURRED, THE DRIVE ATTENTION WILL HAPPEN A LITTLE LATER
6222 ; (ABOUT 3 MILISECONDS) AT WHICH TIME THE DRIVE MUST BE SELECTED
6223 ; TO CHECK THAT DRIVE OFF TRACK SET.
6224 ;
6225 ;*****
6226 ;*****
6227
6228 ; THIS DRIVE LOGIC MAY CHANGE!!! IF IT DOES THIS
6229 ; TEST MUST BE ALTERED.
6230
6231 025326 005037 001662 CLR INTSET ;CLEAR INT FLAG
6232 025332 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6233 025334 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6234
6235 025336 104423 TWAT16 ;WAIT FOR INTERRUPT FROM DRIVE
6236 ;THAT SIGNALS DRIVE OFF TRACK ERROR
6237 025340 104002 ERROR 2
6238
6239 025342 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6240 025346 000101 SELDRV ;SELDV
6241 025350 000000 0 ;0 WORDS

```

```

6242 025352 000000 0 ;0 IS BUFF ADDRESS
6243 025354 000 .BYTE 0 ;SECTOR 0
6244 025355 000 .BYTE 0 ;TRACK 0
6245 025356 000000 0 ;CYLINDER 0
6246 025360 005037 0C1616 CLR L.MR1 ;RESET DIAG MODE
6247
6248 025364 104417 TLOADRK ;LOAD RK REGS
6249 025366 104423 TWAT16 ;WAIT FOR INTERRUPT
6250 025370 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6251
6252 025372 104422 TCHKWE ;CHECK OPERATION WITH ERROR EXPECTED
6253 025374 000400 DROTERR ;DRIVE OFF TRACK
6254 025376 000000 0
6255 025400 000000 0
6256 025402 104004 ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
6257
6258 *****
6259 *TEST 103 FILE UNSAFE
6260
6261 * ISSUE A SUBSYSTEM CLEAR. ISSUE A RECLAIBRATE. ISSUE
6262 * A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
6263 * FORMAT. DO A SELECT COMMAND IN 26 SECTOR FORMAT.
6264 * PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
6265 * HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
6266 * FORMAT. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
6267 * SIMULATE INDEX PULSE. TURN OFF DIAGNOSTIC MODE. FILE
6268 * UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
6269 * THROUGH SECTOR PULSE. REPEAT FOR ALL AVAILIABLE DRIVES.
6270 *****
6271 025404 000004 ST103: SCOPE
6272 025406 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
6273 025414 104416 TSSINIT ;CLEAR SUBSYSTEM
6274 025416 104003 ERROR 3 ;BAD INIT ERROR
6275
6276 025420 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6277 025424 000113 RECAL ;RECAL
6278 025426 000000 0 ;0 WORDS
6279 025430 000000 0 ;0 IS BUFF ADDRESS
6280 025432 000 .BYTE 0 ;SECTOR 0
6281 025433 000 .BYTE 0 ;TRACK 0
6282 025434 000000 0 ;CYLINDER 0
6283
6284 025436 104417 TLOADRK ;LOAD RK REGS
6285 025440 104423 TWAT16 ;WAIT FOR INTERRUPT
6286 025442 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6287
6288 025444 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6289 025446 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6290
6291 025450 005037 001662 CLR INTSET ;CLEAR INT FLAG
6292 025454 104437 TWAT8S ;WAIT FOR SECOND INT
6293 025456 104002 ERROR 2
6294
6295 025460 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6296 025462 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6297

```



```

6298 025464 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6299 025470 000105 CLEAR ;CLEAR
6300 025472 000000 0 ;0 WORDS
6301 025474 000000 0 ;0 IS BUFF ADDRESS
6302 025476 000 .BYTE 0 ;SECTOR 0
6303 025477 000 .BYTE 0 ;TRACK 0
6304 025500 000000 0 ;CYLINDER 0
6305
6306 025502 104417 TLOADRK ;LOAD RK REGS
6307 025504 104423 TWAT16 ;WAIT FOR INTERRUPT
6308 025506 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6309
6310 025510 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6311 025512 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6312
6313 025514 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6314 025520 010125 RDHEAD!CFMT ;RDHEAD!CFMT
6315 025522 000000 0 ;0 WORDS
6316 025524 000000 0 ;0 IS BUFF ADDRESS
6317 025526 000 .BYTE 0 ;SECTOR 0
6318 025527 000 .BYTE 0 ;TRACK 0
6319 025530 000000 0 ;CYLINDER 0
6320
6321 025532 104417 TLOADRK ;LOAD RK REGS
6322 025534 104424 TWAT32 ;WAIT FOR INTERRUPT
6323 025536 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6324
6325 025540 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6326 025542 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6327
6328 025544 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6329 025550 000101 SELDRV ;SELDV
6330 025552 000000 0 ;0 WORDS
6331 025554 000000 0 ;0 IS BUFF ADDRESS
6332 025556 000 .BYTE 0 ;SECTOR 0
6333 025557 000 .BYTE 0 ;TRACK 0
6334 025560 000000 0 ;CYLINDER 0
6335
6336 025562 104417 TLOADRK ;LOAD RK REGS
6337 025564 104423 TWAT16 ;WAIT FOR INTERRUPT
6338 025566 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6339
6340 025570 004437 033324 JSR R4,LRLOAD ;LOAD "L" REGS
6341 025574 000127 WRHEAD ;WRHEAD
6342 025576 177777 -1 ;-1 WORDS
6343 025600 060446 OBUFF ;OBUFF IS BUFF ADDRESS
6344 025602 000 .BYTE 0 ;SECTOR 0
6345 025603 000 .BYTE 0 ;TRACK 0
6346 025604 000000 0 ;CYLINDER 0
6347 025606 012737 000040 001616 MOV #DMD,L.MR1 ;SET DIAGNOSTIC-MODE
6348
6349 025614 104417 TLOADRK ;LOAD RK REGS
6350 025616 004437 033600 JSR R4,MCLOCK ;CLOCK THROUGH SEEK AND DRIVE CLEAR
6351 025622 001064 1064
6352
6353 025624 052762 000200 000026 BIS #MIND,RKMR1(R2) ;SET INDEX

```

6354								
6355	025632	004437	033600	JSR	R4,MCLOCK			;CLOCK INDEX
6356	025636	001001		1001				
6357								
6358	025640	042762	000200 000026	BIC	#MIND,RKMR1(R2)			;CLEAR INDEX
6359								
6360	025646	004437	033600	JSR	R4,MCLOCK			;CLOCK CLEAR
6361	025652	001001		1001				
6362								
6363	025654	005062	000026	CLR	RKMR1(R2)			;CLEAR DIAGNOSTIC MODE
6364								
6365	025660	104426		TWAT64				;WAIT FOR INTERRUPT
6366	025662	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
6367								
6368	025664	104421		TCHKOP				;CHECK OPERATION FOR ANY ERRORS
6369	025666	104004		ERROR	4 ;OR 5, 6, 7, 10			;REPORT ALL ERRORS
6370								
6371	025670	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
6372	025674	000101		SELDRV				;SELDRV
6373	025676	000000		0				;0 WORDS
6374	025700	000000		0				;0 IS BUFF ADDRESS
6375	025702	000		.BYTE	0			;SECTOR 0
6376	025703	000		.BYTE	0			;TRACK 0
6377	025704	000000		0				;CYLINDER 0
6378								
6379	025706	005037	001616	CLR	L.MR1			;CLEAR DIAG MODE
6380								
6381	025712	104417		TLOADRK				;LOAD RK REGS
6382	025714	104423		TWAT16				;WAIT FOR INTERRUPT
6383	025716	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
6384								
6385	025720	104422		TCHKWE				;CHECK OPERATION WITH EXPECTED ERROR
6386	025722	040400		UNSERR!DROTERR				;UNSAFE AND DRIVE OFF TRACK
6387	025724	000000		0				
6388	025726	000000		0				
6389	025730	104004		ERROR	4; OR 5,6,7			;REPORT ANY DISCREPANCIES
6390								
6391	025732	104416		TSSINIT				;CLEAR SUBSYSTEM
6392	025734	104003		ERROR	3			;BAD INIT ERROR
6393								
6394	025736	004437	033324	JSR	R4,LRLOAD			;LOAD "L" REGS
6395	025742	000101		SELDRV				;SELDRV
6396	025744	000000		0				;0 WORDS
6397	025746	000000		0				;0 IS BUFF ADDRESS
6398	025750	000		.BYTE	0			;SECTOR 0
6399	025751	000		.BYTE	0			;TRACK 0
6400	025752	000000		0				;CYLINDER 0
6401								
6402	025754	012737	000001 001616	MOV	#1,L.MR1			;SET MESSAGE SELECT ONE
6403								
6404	025762							
6405	025762	104417		TLOADRK				;LOAD RK REGS
6406	025764	104423		TWAT16				;WAIT FOR INTERRUPT
6407	025766	104002		ERROR	2			;TO SLOW/NOT COMPLETE ERROR
6408								
6409	025770	104421		TCHKOP				;CHECK OPERATION FOR ANY ERRORS

15:

```

6410 025772 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6411
6412 025774 032737 000040 001574 BIT #S.HDHM,T.MR2 ;TEST IF HEADS HOME
6413 026002 001767          BEQ 1$
6414
6415 026004 104416          TSSINIT ;CLEAR SUBSYSTEM
6416 026006 104003          ERROR 3 ;BAD INIT ERROR
6417
6418 026010 005037 001662 CLR INTSET ;CLEAR INT FLAG
6419 026014 104434          TWAT159 ;WAIT FOR APPROX 160 MS
6420 026016 000240          NOP ;DON'T CARE ERROR RETURN
6421
6422 026020 104416          TSSINIT ;CLEAR SUBSYSTEM
6423 026022 104003          ERROR 3 ;BAD INIT ERROR
6424
6425 026024 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
6426
6427 026032 104437          TWAT8S ;WAIT FOR SECOND INTERRUPT
6428 026034 104002          ERROR 2
6429
6430 026036 005037 001616 CLR L.MR1 ;CLEAR MR1
6431
6432 026042 004437 033324 JSR R4,LRLoad ;LOAD "L" REGS
6433 026046 000105          CLEAR ;CLEAR
6434 026050 000000          0 ;0 WORDS
6435 026052 000000          0 ;0 IS BUFF ADDRESS
6436 026054 000          .BYTE 0 ;SECTOR 0
6437 026055 000          .BYTE 0 ;TRACK 0
6438 026056 000000          0 ;CYLINDER 0
6439
6440 026060 104417          TLOADRK ;LOAD RK REGS
6441 026062 104423          TWAT16 ;WAIT FOR INTERRUPT
6442 026064 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6443
6444 026066 104421          TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6445 026070 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6446
6447 026072 004437 037526 JSR R4,GENCOM ;BUILD HEADERS
6448 026076 001200          1200
6449
6450 026100 004437 033324 JSR R4,LRLoad ;LOAD "L" REGS
6451 026104 000127          WRHEAD ;WRHEAD
6452 026106 177676          -102 ;-102 WORDS
6453 026110 060446          OBUFF ;OBUFF IS BUFF ADDRESS
6454 026112 000          .BYTE 0 ;SECTOR 0
6455 026113 000          .BYTE 0 ;TRACK 0
6456 026114 000000          0 ;CYLINDER 0
6457
6458 026116 104417          TLOADRK ;LOAD RK REGS
6459 026120 104426          TWAT64 ;WAIT FOR INTERRUPT
6460 026122 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6461
6462 026124 104421          TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6463 026126 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6464
6465

```

;;\*\*\*\*\*

```

6466      ;*TEST 104      DUMMY TEST FOR PREVIOUS TEST EXIT
6467      ;*
6468      ;* THIS TEST IS PRESENT TO MAKE $SWO8TB TABLE HAVE AN ENTRY
6469      ;* WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
6470      ;* IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST.
6471      ;* IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
6472      ;* "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
6473      ;*
6474      ;* IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
6475      ;* TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
6476      ;* FROM THE DRIVE COMING READY AT A LATER TIME.
6477      ;* *****
6477      026130 000004      †ST104: SCOPE
6478      026132 012737 000001 001262      MOV #1,$TIMES      ;;DO 1 ITERATION
6479
6480      026140 104416      TSSINIT      ;CLEAR SUBSYSTEM
6481      026142 104003      ERROR 3      ;BAD INIT ERROR
6482
6483      026144 013762 001626 000010      MOV DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
6484      026152 012762 000001 000000 1$: MOV #1,RKS1(R2) ;SELECT THE DRIVE
6485      026160 032762 000200 000012      BIT #DRDY,RKDS(R2) ;TEST IF DRIVE READY
6486      026166 001771      BEQ 1$      ;NO LOOP
6487
6488      026170 104416      TSSINIT      ;CLEAR SUBSYSTEM
6489      026172 104003      ERROR 3      ;BAD INIT ERROR
6490

```

```

6491          .SBTTL **MULTI-DRIVE OPERATIONS
6492 026174 000004 NEWDRV: SCOPE
6493 026176 012737 000001 001262 MOV #1,$TIMES ;DO ONLY ONCE
6494 026204 032737 000200 001630 BIT #BIT7,DRVBIT ;WERE WE TESTING DRIVE 7?
6495 026212 001022 BNE 3$ ;YES-SKIP
6496
6497 026214 005237 001626 1$: INC DRVNUM ;BUMP TO NEXT SEQUENTIAL ADDRESS
6498 026220 006337 001630 ASL DRVBIT ;BUMP DRIVEBIT TO THAT POSITION
6499 026224 033737 001630 001354 BIT DRVBIT,$DEVM ;IS THIS DRIVE TO BE TESTED?
6500 026232 001005 BNE 2$ ;YES-EXIT
6501 026234 032737 000400 001630 BIT #BIT8,DRVBIT ;ALL DRIVES TESTED?
6502 026242 001006 BNE 3$ ;YES-EXIT
6503 026244 000763 BR 1$ ;ELSE CHECK NEXT DRIVE AVAILABLE
6504
6505 026246 112737 000004 001102 2$: MOVB #4,$TSTNM ;SET TEST NUMBER FOR REPORTS
6506 026254 000137 004452 JMP TSTLUP ;GO TO TEST LOOP TO CHECK THIS DRIVE
6507 026260 005037 001630 3$: CLR DRVBIT ;CLEAR DRIVE BIT
6508 026264 005037 001626 CLR DRVNUM ;CLEAR DRIVE NUMBER
6509
6510          ;*****
6511          ;*TEST 105 RESET ATTENTIONS WITH UNIBUS INIT
6512          ;*
6513          ;* DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
6514          ;* ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.
6515          ;*
6516          ;*****
6517 026270 000004 TST105: SCOPE
6518 026272 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
6519 026300 005000 CLR RO ;CLEAR DRIVE POSITION COUNTER
6520 026302 012701 000001 MOV #1,R1 ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
6521 026306 013703 001354 MOV $DEVM,R3 ;GET DEVICE MAP
6522 026312 104416 TSSINIT ;CLEAR SUBSYSTEM
6523 026314 104003 ERROR 3 ;BAD INIT ERROR
6524 026316 030103 1$: BIT R1,R3 ;TEST IF THIS DRIVE AVAILABLE
6525 026320 001006 BNE 2$ ;YES-SKIP TO SEEK
6526 026322 006301 3$: ASL R1 ;SHIFT DRIVE SELECT BIT
6527 026324 005200 INC RO ;BUMP DRIVE POSITION COUNTER
6528 026326 032700 000400 BIT #BIT8,RO ;ALL DRIVE POSITIONS CHECKED
6529 026332 001371 BNE 1$ ;NO-LOOP
6530 026334 000417 BR 4$ ;SKIP TO RESET
6531
6532 026336 010037 001610 2$: MOV RO,L.CS2 ;LOAD DRIVE NUMBER
6533 026342 012737 000113 001600 MOV #RECAL,L.CS1 ;LOAD RECALIBRATE
6534
6535 026350 104417 TLOADRK ;LOAD RK REGS
6536 026352 104425 TWAT48 ;WAIT FOR INTERRUPT
6537 026354 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6538
6539 026356 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
6540 026362 104437 TWAT8S ;WAIT FOR SECOND INTERRUPT
6541 026364 104002 ERROR 2
6542
6543 026366 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6544 026370 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6545
6546 026372 000753 BR 3$ ;LOOP FOR NEXT DRIVE
  
```

```

6547 026374 000005      4$:  RESET                      ;UNIBUS RESET
6548
6549 026376 012701 000031      MOV  #25.,R1                ;DO A SHORT DELAY
6550 026402 005301      5$:  DEC  R1
6551 026404 001376      BNE  5$
6552 026406 004737 032434      JSR  PC,OPTTST              ;SET UP OPTIONS
6553
6554 026412 104420      TGETRK                      ;GET RK611 REGS
6555
6556 026414 105737 001557      TSTB T.ASOF+1              ;ALL ATTENTION RESET?
6557 026420 001407      BEQ  6$                    ;YES-SKIP
6558
6559 026422 012737 053262 001470      MOV  #EMDA,EM12N           ;"DRIVE ATT NOT RESET RESULT OF
6560 026430 012737 053406 055544      MOV  #EMUR,DF011A         ;UNIBUS RESET"
6561 026436 104012      ERROR 12
6562
6563 026440      6$:
6564      ;*****
6565      ;*TEST 106      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
6566      ;*
6567      ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
6568      ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
6569      ;*      RESET.
6570      ;*
6571      ;*****
6572 026440 000004      TST106: SCOPE
6573 026442 012737 000062 001262      MOV  #50.,$TIMES          ;:DO 50. ITERATIONS
6574 026450 005000      CLR  R0                    ;:CLEAR DRIVE POSITION COUNTER
6575 026452 012701 000001      MOV  #1,R1                 ;:PRESET TO TEST POSITION 0
6576 026456 013703 001354      MOV  $DEVN,R3              ;:CUT DEVICE MAP
6577 026462 104416      TSSINIT                    ;:CLEAR SUBSYSTEM
6578 026464 104003      ERROR 3                    ;:BAD INIT ERROR
6579 026466 030103      1$:  BIT  R1,R3               ;:THIS DRIVE AVAILABLE?
6580 026470 001006      BNE  2$                    ;:YES-SKIP TO SEEK
6581 026472 006301      3$:  ASL  R1                 ;:SHIFT TO NEXT DRIVE POSITION
6582 026474 005200      INC  R0                    ;:DUMP POSITION COUNTER
6583 026476 032701 000400      BIT  #BIT8,R1              ;:ALL POSITIONS CHECKED
6584 026502 001371      BNE  1$                    ;:NO-LOOP
6585 026504 000417      BR   4$                    ;:YES-SKIP TO CLEAR
6586
6587 026506 010037 001610 2$:  MOV  R0,L.CS2              ;:LOAD DRIVE NUMBER
6588 026512 012737 000113 001600      MOV  #RECAL,L.CS1         ;:LOAD RECALIBRATE
6589 026520 104417      TLOADRK                    ;:LOAD RK REGS
6590 026522 104423      TWAT16                     ;:WAIT FOR INTERRUPT
6591 026524 104002      ERROR 2                    ;:TO SLOW/NOT COMPLETE ERROR
6592
6593 026526 005037 001662      CLR  INTSET                 ;:CLEAR INT FLAG
6594 026532 104437      TWAT8S                      ;:WAIT FOR SECOND INT
6595 026534 104002      ERROR 2
6596
6597 026536 104421      TCHKOP                      ;:CHECK OPERATION FOR ANY ERRORS
6598 026540 104004      ERROR 4 ;OR 5, 6, 7      ;:REPORT ALL ERRORS
6599
6600 026542 000753      BR   3$                    ;:LOOP FOR NEXT DRIVE
6601
6602 026544 052762 000040 000010 4$:  BIS  #SCLR,RKCS2(R2)      ;DO SUBSYSTEM CLEAR

```

```

6603 026552 012701 000031          MOV    #25.,R1      ;DO A SHORT DELAY
6604 026556 005301          SS:   DEC    R1
6605 026560 001376          BNE    SS
6606
6607 026562 104420          TGETRK          ;GET RK611 REGS
6608
6609 026564 105737 001557          TSTB   T,ASOF+1   ;TEST ALL ATTENTION RESET
6610 026570 001407          BEQ    SS         ;YES-SKIP
6611
6612 026572 012737 053262 001470          MOV    #EMDA,EM12N ;"DRIVE ATT NOT RESET AS RESULT OF
6613 026600 012737 053476 055544          MOV    #EMSCLR,DF011A ;SUBSYSTEM CLEAR"
6614 026606 104012          ERROR  12
6615
6616 026610          SS:
6617          ;*****
6618          ;*TEST 107          SVAL AND ATTENTION FROM OTHER DRIVE
6619          ;*
6620          ;*          DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT
6621          ;*          ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID
6622          ;*          IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
6623          ;*          MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
6624          ;*          CHANGE REMAINS RESET.
6625          ;*
6626          ;*          REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
6627          ;*
6628          ;*          NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST
6629          ;*          TWO DRIVES ARE AVAILABLE.
6630          ;*
6631          ;*****
6632 026610 000004          TST107: SCOPE
6633 026612 012737 000024 001262          MOV    #20.,STIMES ;:DO 20. ITERATIONS
6634 026620 013746 001354          MOV    $DEVN,-(SP) ;:PUT DEVICE MAP ON STACK
6635 026624 004437 033524          JSR    R4,BITCNT  ;:COUNT NUMBER OF BITS(# OF DRIVES)
6636 026630 022627 000001          CMP    (SP)+,#1   ;:COMPARE TO 1
6637 026634 101007          BHI    SS         ;:SKIP IF MORE THAN 1
6638 026636 005737 001304          TST    $PASS      ;:CHECK IF PASS 0
6639 026642 001002          BNE    SS         ;:NO-SKIP
6640
6641 026644 104401 047540          TYPE   OPRO18     ;:"OVERLAPPED OPERATION BYPASSED"
6642 026650 000137 030064          SS:   JMP    $EOP    ;:GET OUT.
6643
6644 026654 012737 177777 001224          SS:   MOV    #-1,$TMP1 ;:SET LOOP CONTROL FLAG
6645 026662 013705 001354          MOV    $DEVN,R5   ;:GET DEVICE MAP
6646 026666 005000          CLR    R0         ;:CLEAR FOR DRIVE #A
6647 026670 005001          CLR    R1         ;:CLEAR FOR DRIVE #B
6648 026672 012703 000001          MOV    #1,R3      ;:SET DRIVE POSITION A
6649 026676 012704 000001          MOV    #1,R4      ;:SET DRIVE POSITION B
6650 026702 012737 026712 001110          MOV    #3,$LPERR  ;:SET LOCAL LOOP ON ERROR
6651 026710 000477          BR     11$        ;:GO SET UP POINTERS
6652
6653 026712          SS:
6654 026712 104416          TSSINIT
6655 026714 104003          ERROR  3         ;:CLEAR SUBSYSTEM
6656          ;:BAD INIT ERROR
6657 026716 010037 001610          MOV    R0,L.C52   ;:LOAD DRIVE A
6658 026722 010037 001626          MOV    R0,DRVNUM  ;:LOAD FOR REPORT

```

```

6659 026726 012737 000113 001600      MOV      #RECAL,L.CS1      ;LOAD RECAL
6660
6661 026734 104417      TLOADRK                    ;LOAD RK REGS
6662 026736 104423      TWAT16                      ;WAIT FOR INTERRUPT
6663 026740 104002      ERROR      2              ;TO SLOW/NOT COMPLETE ERROR
6664 026742 104421      TCHKOP                      ;CHECK OPERATION FOR ANY ERRORS
6665 026744 104004      ERROR      4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6666
6667 026746 005037 001662      CLR      INTSET            ;CLEAR INT FLAG
6668
6669 026752 010137 001610      MOV      R1,L.CS2          ;LOAD DRIVE B
6670 026756 010137 001626      MOV      R1,DRVNUM         ;LOAD FOR REPORT
6671 026762 012737 000101 001600      MOV      #SELDRV,L.CS1    ;LOAD DRIVE SELECT
6672
6673 026770 104417      TLOADRK                    ;LOAD RK REGS
6674 026772 104423      TWAT16                      ;WAIT FOR INTERRUPT
6675 026774 104002      ERROR      2              ;TO SLOW/NOT COMPLETE ERROR
6676
6677 026776 104421      TCHKOP                      ;CHECK OPERATION FOR ANY ERRORS
6678 027000 104004      ERROR      4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6679
6680 027002 032737 100000 001552      BIT      #SVAL,T.DS        ;CHECK IF STATUS VALID SET
6681 027010 001007      BNE      4$                ;YES - SKIP
6682 027012 012737 053076 001460      MOV      #EMSVAL,EM11N     ;"STATUS VALID NOT SET RESULT OF
6683 027020 012737 045134 055544      MOV      #OPER00,DF011A   ;DRIVE SELECT"
6684 027026 104011      ERROR      11
6685
6686 027030 005037 001662      4$: CLR      INTSET            ;CLEAR INT FLAG
6687 027034 104436      TWAT2S                      ;WAIT FOR SEEK COMPLETE INTERRUPT
6688 027036 000401      BR      44$                ;NONE RECEIVED - SKIP
6689 027040 000406      BR      55$                ;RECEIVED - SKIP
6690
6691 027042 010037 001626      44$: MOV      R0,DRVNUM       ;SET DRIVE FOR REPORT
6692 027046 012737 055133 001372      MOV      #DHD17,DH2N      ;"COMMAND - SELECT AFTER RECAL"
6693 027054 104002      ERROR      2
6694
6695 027056 104420      55$: TGETRK                    ;GET RK REGS
6696 027060 032737 100000 001552      BIT      #SVAL,T.DS        ;TEST IF SVAL STILL SET
6697 027066 001007      BNE      5$                ;YES - SKIP
6698
6699 027070 012737 053076 001510      MOV      #EMSVAL,EM14N     ;"STATUS VALID RESET RESULT OF
6700 027076 012737 055156 055544      MOV      #DHD18,DF011A   ;RECAL COMPLETE ATTENTION AFTER SEL"
6701 027104 104014      ERROR      14
6702
6703 027106 104415      5$: SCOP1                    ;LOCAL LOOP TO 3$
6704
6705 :
6706 :
6707 :
6708 :
6709 :
6710 :
6711 :
6712 :
6713 :
6714 :
  
```

THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION OF DRIVES AVAILABLE. THE FIRST PASS OF THE PROGRAM WILL USE THE LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS B. THE SECOND PASS SWAPS DRIVE A & B. THE THIRD PASS USES THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE AS A. THE FORTH PASS SWAPS A & B AGAIN. THIS CONTINUES UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER DRIVE.

THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE



```

6715      :      THIRD HIGHEST AS B.  THEY ARE SWAPPED ON THE NEXT PASS.
6716      :
6717      :
6718      :      THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
6719      :      CHECKED.
6720 027110 005237 001224 11$:  INC      $TMP1      ;INCREMENT PASS CONTROL
6721 027114 001024          BNE      16$      ;SKIP IF NOT ZERO
6722          :                      ;(IT WILL BE ZERO ON THE 1ST PASS)
6723
6724 027116 030305 12$:  BIT      R3,R5      ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
6725 027120 001006          BNE      13$      ;YES-SKIP
6726
6727 027122 005200 22$:  INC      R0          ;BUMP R0 (DRIVE A)
6728 027124 006303          ASL      R3          ;SHIFT DRIVE SELECT BIT ONE POSITION
6729 027126 032703 000400 BIT      #BIT8,R3      ;IF BIT 8 IS SET, ALL DRIVES HAVE
6730 027132 001771          BEQ      12$      ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
6731 027134 000464          BR       50$      ;DONE-EXIT
6732
6733 027136 010001 13$:  MOV      R0,R1      ;SET DRIVE B TO THE SAME AS A
6734 027140 010304          MOV      R3,R4
6735 027142 005201 14$:  INC      R1          ;BUMP R1 (DRIVE B)
6736 027144 006304          ASL      R4          ;SHIFT SELECTOR BIT ONE POSITION
6737 027146 030405          BIT      R4,R5      ;IS THIS DRIVE AVAIL?
6738 027150 001004          BNE      15$      ;YES-SKIP
6739 027152 032704 000400 BIT      #BIT8,R4      ;WERE ALL POSITIONS CHECKED?
6740 027156 001771          BEQ      14$      ;NO-LOOP
6741 027160 000452          BR       50$      ;DONE-EXIT
6742
6743 027162 000137 026712 15$:  JMP      3$          ;GO DO THE TEST ON THE DRIVE A & B
6744          :                      ;CONTAINED IN R0 & R1
6745 027166 032737 000001 001224 16$: BIT      #BIT0,$TMP1      ;IS PASS FLAGS ODD?
6746 027174 001410          BEQ      17$      ;NO-SKIP
6747
6748 027176 010046          MOV      R0,-(SP)      ;
6749 027200 010346          MOV      R3,-(SP)      ;SWAP R0 & R1, R3 & R4
6750 027202 010403          MOV      R4,R3      ;TO EXCHANGE DRIVE A & B
6751 027204 010100          MOV      R1,R0
6752 027206 012604          MOV      (SP)+,R4
6753 027210 012601          MOV      (SP)+,R1
6754 027212 000137 026712 JMP      3$          ;REPEAT TEST ON THIS COMBO.
6755
6756 027216 032737 000002 001224 17$: BIT      #BIT1,$TMP1      ;TEST IF PASS FLAGS AT HALF MODULE 4?
6757 027224 001410          BEQ      19$      ;NO-SKIP TO BUMP DRIVE B
6758 027226 005200 18$:  INC      R0          ;BUMP DRIVE A
6759 027230 006303          ASL      R3          ;SHIFT DRIVE SELECT BIT
6760 027232 030305          BIT      R3,R5      ;AVAILABLE?
6761 027234 001014          BNE      20$      ;YES-SKIP
6762 027236 032703 000400 BIT      #BIT8,R3      ;ALL CHECKED?
6763 027242 001771          BEQ      18$      ;NO-SKIP
6764 027244 000412          BR       21$      ;GO TO NEXT PASS
6765
6766 027246 005201 19$:  INC      R1          ;BUMP DRIVE B
6767 027250 006304          ASL      R4          ;SHIFT DRIVE SELECT BIT
6768 027252 030405          BIT      R4,R5      ;AVAILABLE?
6769 027254 001004          BNE      20$      ;YES-SKIP
6770 027256 032704 000400 BIT      #BIT8,R4      ;ALL CHECKED?

```

E13

```

6771 027262 001771          BEQ      19$          ;NO-LOOP
6772 027264 000404          BR       23$          ;YES-SKIP TO NEXT PASS
6773
6774 027266 000137 026712 20$:  JMP      3$          ;GO TEST THIS COMBO
6775
6776 027272 010100          21$:  MOV      R1,R0       ;SET DRIVE 0 TO LOW POSITION THIS PASS
6777 027274 010403          MOV      R4,R3       ;SET SELECT BITS TO AGREE
6778 027276 005037 001224 23$:  CLR      $TMP1       ;CLEAR PASS FLAGS
6779 027302 000137 027122          JMP      22$          ;GO SET UP A & B
6780 027306
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796 027306 000004          *****
6797 027310 012737 000024 001262 1ST110: SCOPE
6798
6799 027316 012737 177777 001224 2$:  MOV      #-1,$TMP1   ;;DO 20. ITERATIONS
6800 027324 013705 001354          MOV      $DEVN,R5    ;SET LOOP CONTROL FLAG
6801 027330 005000          CLR      R0          ;GET DEVICE MAP
6802 027332 005001          CLR      R1          ;CLEAR FOR DRIVE #A
6803 027334 012703 000001          MOV      #1,R3       ;CLEAR FOR DRIVE #B
6804 027340 012704 000001          MOV      #1,R4       ;SET DRIVE POSITION A
6805 027344 012737 027354 001110          MOV      #3,$LPERR   ;SET DRIVE POSITION B
6806 027352 000545          BR       11$         ;SET LOCAL LOOP ON ERROR
6807 027354          3$:  BR       11$         ;GO SET UP POINTERS
6808 027354 104416          TSSINIT           ;CLEAR SUBSYSTEM
6809 027356 104003          ERROR  3          ;BAD INIT ERROR
6810
6811 027360 010037 001626          MOV      R0,DRVNUM   ;STORE DRIVE FOR REPORT
6812 027364 010037 001610          MOV      R0,L.CS2    ;SETUP DRIVE A TO RECAL
6813 027370 012737 000113 001600          MOV      #RECAL,L.CS1
6814
6815 027376 104417          TLOADRK           ;LOAD RK REGS
6816 027400 104423          TWAT16           ;WAIT FOR INTERRUPT
6817 027402 104002          ERROR  2          ;TO SLOW/NOT COMPLETE ERROR
6818 027404 005037 001662          CLR      INTSET     ;CLEAR INTERRUPT FLAG
6819
6820 027410 104437          TWAT8S           ;WAIT FOR SECOND INTERRUPT
6821 027412 104002          ERROR  2          ;TO SLOW/NOT COMPLETE ERROR
6822
6823 027414 012737 000105 001600          MOV      #CLEAR,L.CS1 ;SET UP TO CLEAR DRIVE
6824 027422 104417          TLOADRK           ;LOAD RK REGS
6825 027424 104423          TWAT16           ;WAIT FOR INTERRUPT
6826 027426 104002          ERROR  2          ;TO SLOW/NOT COMPLETE ERROR

```

```

6827 027430          4$:
6828 027430 104421      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6829 027432 104004      ERROR          4 ;OR 5 6, 7 ;REPORT ALL ERRORS
6830 027434 032737 040000 001540  BIT          #DI,T.CS1 ;TEST IF DI STILL SET
6831 027442 001372      BNE          4$ ;YES - LOOP
6832 027444 010137 001626  MOV          R1,DRVNUM ;STORE DRIVE FOR REPORT
6833 027450 010137 001610  MOV          R1,L.CS2 ;SETUP DRIVE B TO RECAL
6834 027454 012737 000113 001600  MOV          #RECAL,L.CS1
6835
6836 027462 104417      TLOADRK          ;LOAD RK REGS
6837 027464 104423      TWAT16          ;WAIT FOR INTERRUPT
6838 027466 104002      ERROR          2 ;TO SLOW/NOT COMPLETE ERROR
6839 027470 005037 001662  CLR          INTSET ;CLEAR INTERRUPT FLAG
6840 027474 104437      TWAT8S          ;WAIT FOR SECOND INTERRUPT
6841 027476 104002      ERROR          2 ;TO SLOW/NOT COMPLETE ERROR
6842 027500 012737 000105 001600  MOV          #CLEAR,L.CS1 ;SET UP DRIVE CLEAR
6843 027506 104417      TLOADRK          ;LOAD RK REGS
6844 027510 104423      TWAT16          ;WAIT FOR INTERRUPT
6845 027512 104002      ERROR          2 ;TO SLOW/NOT COMPLETE ERROR
6846
6847 027514 104421      5$:          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6848 027516 104004      ERROR          4 ;OR 5 6, 7 ;REPORT ALL ERRORS
6849 027520 032737 040000 001540  BIT          #DI,T.CS1 ;TEST IF DI STILL SET
6850 027526 001372      BNE          5$ ;YES - LOOP
6851
6852 027530 010037 001626  MOV          R0,DRVNUM ;STORE DRIVE FOR REPORT
6853 027534 010037 001610  MOV          R0,L.CS2 ;SETUP DRIVE A TO SEEK
6854 027540 012737 000001 001614  MOV          #1,L.DCYL ;TO CYL 1
6855 027546 012737 000117 001600  MOV          #SEEK,L.CS1
6856
6857 027554 104417      TLOADRK          ;LOAD RK REGS
6858 027556 104423      TWAT16          ;WAIT FOR INTERRUPT
6859 027560 104002      ERROR          2 ;TO SLOW/NOT COMPLETE ERROR
6860
6861 027562 010137 001626  MOV          R1,DRVNUM ;STORE DRIVE FOR REPORT
6862 027566 010137 001610  MOV          R1,L.CS2 ;SETUP DRIVE B TO WRITE DATA
6863 027572 012737 000100 001614  MOV          #100,L.DCYL ;AT CYL 100
6864 027600 012737 177400 001602  MOV          #-400,L.WC ;400 WORDS
6865 027606 012737 060446 001604  MOV          #OBUF,L.BA
6866 027614 012737 000123 001600  MOV          #WRDATA,L.CS1
6867
6868 027622 104417      TLOADRK          ;LOAD RK REGS-DO WRITE
6869
6870 027624 104427      TWAT80          ;WAIT FOR INTERRUPT
6871 027626 104002      ERROR          2 ;TO SLOW/NOT COMPLETE ERROR
6872
6873 027630 104421      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6874 027632 104004      ERROR          4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6875
6876 027634 010037 001626  MOV          R0,DRVNUM ;STORE DRIVE FOR REPORT
6877 027640 130337 001557  BITB         R3,T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS DRIVE A
6878 027644 001007 10$          BNE          10$ ;YES-SKIP
6879 027646 012737 053262 001460  MOV          #EMDA,EM11N ;"DRIVE ATTENTION NOT SET RESULT OF
6880 027654 012737 045246 055544  MOV          #EMSK,DF011A ;SEEK"
6881 027662 104011      ERROR          11
6882

```

```

6883 027664 104415      10$: SCOP1                ;LOCAL LOOP TO 3$
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900 027666 005237 001224      11$: INC      $TMP1        ;INCREMENT PASS CONTROL
6901 027672 001024                BNE      16$            ;SKIP IF NOT ZERO
6902
6903
6904 027674 030305      12$: BIT      R3,R5        ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
6905 027676 001006                BNE      13$            ;YES-SKIP
6906
6907 027700 005200      22$: INC      R0            ;BUMP R0 (DRIVE A)
6908 027702 006303                ASL     R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
6909 027704 032703 000400      BIT     #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
6910 027710 001771                BEQ     12$            ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
6911 027712 000464                BR      50$            ;DONE-EXIT
6912
6913 027714 010001      13$: MOV     R0,R1          ;SET DRIVE B TO THE SAME AS A
6914 027716 010304                MOV     R3,R4
6915 027720 005201      14$: INC      R1            ;BUMP R1 (DRIVE B)
6916 027722 006304                ASL     R4              ;SHIFT SELECTOR BIT ONE POSITION
6917 027724 030405      BIT     R4,R5          ;IS THIS DRIVE AVAIL?
6918 027726 001004                BNE     15$            ;YES-SKIP
6919 027730 032704 000400      BIT     #BIT8,R4        ;WERE ALL POSITIONS CHECKED?
6920 027734 001771                BEQ     14$            ;NO-LOOP
6921 027736 000452                BR      50$            ;DONE-EXIT
6922
6923 027740 000137 027354      15$: JMP      3$            ;GO DO THE TEST ON THE DRIVE A & B
6924
6925 027744 032737 000001 001224 16$: BIT     #BIT0,$TMP1    ;CONTAINED IN R0 & R1
6926 027752 001410                BEQ     17$            ;IS PASS FLAGS ODD?
6927
6928 027754 010046                MOV     R0,-(SP)        ;
6929 027756 010346                MOV     R3,-(SP)        ;SWAP R0 & R1, R3 & R4
6930 027760 010403                MOV     R4,R3          ;TO EXCHANGE DRIVE A & B
6931 027762 010100                MOV     R1,R0
6932 027764 012604                MOV     (SP)+,R4
6933 027766 012601                MOV     (SP)+,R1
6934 027770 000137 027354      JMP     3$            ;REPEAT TEST ON THIS COMBO.
6935
6936 027774 032737 000002 001224 17$: BIT     #BIT1,$TMP1    ;TEST IF PASS FLAGS AT HALF MODULE 4?
6937 030002 001410                BEQ     19$            ;NO-SKIP TO BUMP DRIVE B
6938 030004 005200      18$: INC      R0            ;BUMP DRIVE A

```

# H13

6939	030006	006303		ASL	R3		;SHIFT DRIVE SELECT BIT
6940	030010	030305		BIT	R3,R5		;AVAILABLE?
6941	030012	001014		BNE	20\$		;YES-SKIP
6942	030014	032703	000400	BIT	#BIT8,R3		;ALL CHECKED?
6943	030020	001771		BEQ	18\$		;NO-SKIP
6944	030022	000412		BR	21\$		;GO TO NEXT PASS
6945							
6946	030024	005201		19\$: INC	R1		;BUMP DRIVE B
6947	030026	006304		ASL	R4		;SHIFT DRIVE SELECT BIT
6948	030030	030405		BIT	R4,R5		;AVAILABLE?
6949	030032	001004		BNE	20\$		;YES-SKIP
6950	030034	032704	000400	BIT	#BIT8,R4		;ALL CHECKED?
6951	030040	001771		BEQ	19\$		;NO-LOOP
6952	030042	000404		BR	23\$		;YES-SKIP TO NEXT PASS
6953							
6954	030044	000137	027354	20\$: JMP	3\$		;GO TEST THIS COMBO
6955							
6956	030050	010100		21\$: MOV	R1,R0		;SET DRIVE 0 TO LOW POSITION THIS PASS
6957	030052	010403		MOV	R4,R3		;SET SELECT BITS TO AGREE
6958	030054	005037	001224	23\$: CLR	\$TMP1		;CLEAR PASS FLAGS
6959	030060	000137	027700	JMP	22\$		;GO SET UP A & B
6960	030064			50\$:			

6961  
6962  
6963  
6964  
6965  
6966  
6967  
6968  
6969 030064  
6970 030064 000004  
6971 030066 005037 001102  
6972 030072 005037 001262  
6973 030076 005237 001304  
6974 030102 042737 100000 001304  
6975 030110 005327  
6976 030112 000001  
6977 030114 003022  
6978 030116 012737  
6979 030120 000001  
6980 030122 030112  
6981 030124 104401 030171  
6982 030130 013746 001304  
6983 030134 104405  
6984 030136 104401 030156  
6985 030142 013700 000042  
6986 030146 001405  
6987 030150 000005  
6988 030152 004710  
6989 030154 000240  
6990 030156 000240  
6991 030160 000240  
6992 030162  
6993 030162 000137  
6994 030164 003076  
6995 030166 377 377 000  
6996 030171 015 042412 042116  
6997 030176 050040 051501 020123  
6998 030204 000043  
6999  
7000  
7001  
7002  
7003  
7004  
7005  
7006  
7007  
7008  
7009  
7010  
7011  
7012  
7013  
7014  
7015 030206 010046  
7016 030210 010146

.SBTTL END OF PASS ROUTINE

```

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

```

```

$EOP:
SCOPE
CLR $STNM ;; ZERO THE TEST NUMBER
CLR $TIMES ;; ZERO THE NUMBER OF ITERATIONS
INC $PASS ;; INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;; LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;; YES
MOV (PC)+,a(PC)+ ;; RESTORE COUNTER
$ENDCT: .WORD 1
TYPE $SENDMG ;; TYPE "END PASS #"
MOV $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
TYPDS ;; GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $ENULL ;; TYPE A NULL CHARACTER
$GET42: MOV a#42,RO ;; GET MONITOR ADDRESS
BEQ $DOAGN ;; BRANCH IF NO MONITOR
RESET ;; CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ;; GO TO MONITOR
NOP ;; SAVE ROOM
NOP ;; FOR
NOP ;; ACT11
$DOAGN: JMP a(PC)+ ;; RETURN
$RTNAD: .WORD TST1
$ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
$ENDMG: .ASCIZ <15><12>/END PASS #/

```

.SBTTL ROUTINE TO SIZE MEMORY

```

*****
; CALL:
; JSR PC,$SIZE
; RETURN
; $LSTAD WILL CONTAIN:
; WITH KT11 OPTION -- LAST VIRTUAL ADDRESS OF THE LAST BANK
; WITHOUT KT11 OPTION -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
; $LSTBK WILL CONTAIN THE LAST BANK AS A SAF
; $KT11 IS THE MEMORY MANAGEMENT KEY
; $BIT07 = 0 DON'T USE MEMORY MANAGEMENT
; MUST BE SETUP BEFORE THE CALL
; $BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
; DETERMINED BY ROUTINE
$SIZE: MOV RO,-(SP) ;; SAVE RO ON THE STACK
MOV R1,-(SP) ;; SAVE R1 ON THE STACK

```

7017	030212	010246				MOV	R2,-(SP)	::SAVE R2 ON THE STACK
7018	030214	010346				MOV	R3,-(SP)	::SAVE R3 ON THE STACK
7019	030216	013746	000004			MOV	@#ERRVEC,-(SP)	::SAVE PRESENT ERROR VECTOR PS & PC
7020	030222	013746	000006			MOV	@#ERRVEC+2,-(SP)	
7021	030226	010600				MOV	SP,RO	::SAVE THE STACK POINTER
7022						;;SET THE ERRVEC PS TO THE PRESENT PS		
7023	030230	104400				TRAP		::PUSH OLD PSW AND PC ON STACK
7024	030232	012637	000006			MOV	(SP)+,@#ERRVEC+2	::SAVE THE PSW IN @#ERRVEC+2
7025	030236	012701	003776			MOV	#3776,R1	::SETUP ADDRESS
7026	030242	105727				TSTB	(PC)+	::USE MEMORY MANAGEMENT?
7027	030244	000200			SKT11:	.WORD	200	::SET TO USE MEMORY MANAGEMENT
7028	030246	100062				BPL	SCORE	::BR IF NO
7029	030250	012737	030406	000004		MOV	#SKTNEX,@#ERRVEC	::SET FOR TIMEOUT
7030	030256	005737	177572			TST	@#SR0	::KT11 ARE YOU THERE?
7031	030262	052737	100000	030244		BIS	#100000,SKT11	::YES--SET KT11 KEY
7032	030270	005046				CLR	-(SP)	::INITIALIZE FOR "PAR" LOADING
7033	030272	012702	172340			MOV	#KIPAR0,R2	::ADDRESS OF FIRST "PAR"
7034	030276	012703	000010			MOV	#108,R3	::LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
7035	030302	012762	077406	177740	1\$:	MOV	#77406,-40(R2)	::PDR = 4K, UP, READ/WRITE
7036	030310	011622				MOV	(SP),(R2)+	::LOAD "PAR"
7037	030312	062716	000200			ADD	#200,(SP)	::UPDATE FOR NEXT "PAR"
7038	030316	077307				SOB	R3,1\$	::LOOP UNTIL ALL EIGHT ARE LOADED
7039	030320	012742	177600			MOV	#177600,-(R2)	::SETUP KIPAR7 FOR I/O
7040	030324	005042				CLR	-(R2)	::SETUP KIPAR6 FOR TESTING
7041	030326	012737	030344	000004		MOV	#2\$,@#ERRVEC	::CATCH TIMEOUT IF NO SR3
7042	030334	012737	000020	172516		MOV	#20,@#SR3	::ENABLE 22 BIT MODE
7043	030342	000401				BR	3\$	::THIS PDP-11 HAS A SR3 REGISTER
7044	030344	022626			2\$:	CMP	(SP)+,(SP)+	::CLEAN OFF THE STACK--NO SR3
7045	030346	005237	177572		3\$:	INC	@#SR0	::TURN ON MEMORY MANAGEMENT
7046	030352	012737	030376	000004		MOV	#SKTOUT,@#ERRVEC	::SET FOR TIME OUT
7047	030360	005737	143776		4\$:	TST	@#143776	::TRAP ON NON-EX-MEM
7048	030364	062712	000040			ADD	#40,(R2)	::MAKE A 1K STEP
7049	030370	023712	172356			CMP	@#KIPAR7,(R2)	::LAST ONE?
7050	030374	101371				BHI	4\$	::NO--TRY IT
7051	030376	011202			SKTOUT:	MOV	(R2),R2	::GET LAST BANK+1
7052	030400	005037	177572			CLR	@#SR0	::TURN OFF MEMORY MANAGEMENT
7053	030404	000401				BR	SSIZEX	
7054	030406	042737	100000	030244	SKTNEX:	BIC	#100000,SKT11	::KT11 NON-EXISTENT
7055	030414	012737	030444	000004	SCORE:	MOV	#SCROUT,@#ERRVEC	::SET FOR TIMEOUT
7056	030422	005002				CLR	R2	::SET UP BANK
7057	030424	062701	004000		1\$:	ADD	#4000,R1	::INCREMENT BY 1K
7058	030430	062702	000040			ADD	#40,R2	::1K STEP
7059	030434	005711				TST	(R1)	::TRAP ON TIME OUT
7060	030436	022701	177776			CMP	#177776,R1	::LAST ONE
7061	030442	001370				BNE	1\$	::NO--TRY AGAIN
7062	030444	162701	004000		SCROUT:	SUB	#4000,R1	
7063	030450	162702	000040		SSIZEX:	SUB	#40,R2	::DROP BACK
7064	030454	010006				MOV	RO,SP	::RESTORE THE STACK
7065	030456	012637	000006			MOV	(SP)+,@#ERRVEC+2	::RESTORE ERROR VECTOR
7066	030462	012637	000004			MOV	(SP)+,@#ERRVEC	
7067	030466	010137	030510			MOV	R1,\$LSTAD	::LAST ADDRESS
7068	030472	010237	030512			MOV	R2,\$LSTBK	::LAST BANK
7069	030476	012603				MOV	(SP)+,R3	::RESTORE R3
7070	030500	012602				MOV	(SP)+,R2	::RESTORE R2
7071	030502	012601				MOV	(SP)+,R1	::RESTORE R1
7072	030504	012600				MOV	(SP)+,RO	::RESTORE RO

```

7073 030506 000207
7074 030510 000000
7075 030512 000000
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090 030514
7091 030514 104407
7092
7093
7094 030516 021627 001002
7095 030522 101002
7096 030524 000137 031530
7097 030530 032777 040000 150402 1$:
7098 030536 001131
7099
7100 030540 000416
7101
7102 030542 013746 000004
7103 030546 012737 030566 000004
7104 030554 005737 177060
7105 030560 012637 000004
7106 030564 000500
7107 030566 022626 5$:
7108 030570 012637 000004
7109 030574 000440
7110 030576 6$:;
7111 030576 032777 000400 150334
7112 030604 001421
7113 030606 005046
7114 030610 117716 150324
7115 030614 001414
7116 030616 022716 000110
7117 030622 002411
7118 030624 011637 001102
7119 030630 005316
7120 030632 006316
7121 030634 062716 031040
7122 030640 013637 001106
7123 030644 000466
7124 030646 005726 8$:
7125 030650 105737 001103 2$:
7126 030654 001421
7127 030656 123737 001115 001103
7128 030664 101015

```

```

RTS PC
$LSTAD: .WORD 0 ;;CONTAINS THE LAST ADDRESS
$LSTBK: .WORD 0 ;;CONTAINS THE LAST BANK
.SBTTL SCOPE HANDLER ROUTINE

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

$SCOPE:
CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
;;OTHERWISE CONTINUE
CMP (SP),#1002 ;;UNEXPECTED TRAP OR INTERRUPT
BHI 1$ ;;ARE TRAPPED HERE VIA IOT
JMP $ERROR ;;GO PROCESS UNEXPECTED TRAP
1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
BNE $OVER ;;YES IF SW14=1
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
;;THIS INSTRUCTION TO A "NOP" (NOP=240)
MOV @#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
MOV #5$, @#ERRVEC ;;SET FOR TIMEOUT
TST @#177060 ;;TIME OUT ON XOR?
MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR
BR $SVLAD ;;GO TO THE NEXT TEST
5$: CMP (SP)+, (SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR
BR 7$ ;;LOOP ON THE PRESENT TEST
6$; *****END OF CODE FOR THE XOR TESTER*****
BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
BEQ 2$ ;;BR IF NO
CLR -(SP) ;;CLEAR A TEMP. LOCATION
MOVB @SWR, (SP) ;;PICKUP THE DESIRED TEST NUMBER
BEQ 8$ ;;BRANCH IF BAD TEST NUMBER IN SWR
CMP #110, (SP) ;;CHECK THE NUMBER IN THE SWR
BLT 8$ ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
MOV (SP), $STNM ;;UPDATE THE TEST NUMBER
DEC (SP) ;;BACKUP BY ONE
ASL (SP) ;;SCALE THE TEST NUMBER AS AN INDEX
ADD #$$SW08TBL, (SP) ;;FORM THE ADDRESS OF TEST POINTER
MOV @($SP)+, $LPADR ;;SET LOOP ADDRESS TO DESIRED TEST
BR $OVER ;;GO LOOP ON THE TEST
8$: TST (SP)+ ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
BEQ 3$ ;;BR IF NO
CMPB $ERMAX, $ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
BHI 3$ ;;BR IF NO

```



```

7129 030666 032777 001000 150244      BIT      #BIT09, @SWR      ;; LOOP ON ERROR?
7130 030674 001404                      BEQ      4$          ;; BR IF NO
7131 030676 013737 001110 001106 7$:   MOV      $LPERR, $LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
7132 030704 000446                      BR       $OVER
7133 030706 105037 001103          4$:   CLR     $ERFLG      ;; ZERO THE ERROR FLAG
7134 030712 005037 001262          CLR     $TIMES      ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
7135 030716 000415                      BR       1$          ;; ESCAPE TO THE NEXT TEST
7136 030720 032777 004000 150212 3$:   BIT     #BIT11, @SWR ;; INHIBIT ITERATIONS?
7137 030726 001011                      BNE     1$          ;; BR IF YES
7138 030730 005737 001304          TST     $PASS       ;; IF FIRST PASS OF PROGRAM
7139 030734 001406                      BEQ     1$          ;; INHIBIT ITERATIONS
7140 030736 005237 001104          INC     $ICNT       ;; INCREMENT ITERATION COUNT
7141 030742 023737 001262 001104      CMP     $TIMES, $ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
7142 030750 002024                      BGE     $OVER       ;; BR IF MORE ITERATION REQUIRED
7143 030752 012737 000001 001104 1$:   MOV     #1, $ICNT   ;; REINITIALIZE THE ITERATION COUNTER
7144 030760 013737 031036 001262      MOV     $MXCNT, $TIMES ;; SET NUMBER OF ITERATIONS TO DO
7145 030766 105237 001102          $SVLAD: INCB    $TSTNM      ;; COUNT TEST NUMBERS
7146 030772 113737 001102 001302      MOV     $TSTNM, $TESTN ;; SET TEST NUMBER IN APT MAILBOX
7147 031000 011637 001106          MOV     (SP), $LPADR ;; SAVE SCOPE LOOP ADDRESS
7148 031004 011637 001110          MOV     (SP), $LPERR ;; SAVE ERROR LOOP ADDRESS
7149 031010 005037 001264          CLR     $ESCAPE     ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
7150 031014 112737 000001 001115      MOV     #1, $ERMAX  ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7151 031022 013777 001102 150112 $OVER: MOV     $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER
7152 031030 013716 001106          MOV     $LPADR, (SP) ;; FUDGE RETURN ADDRESS
7153 031034 000002                      RTI
7154 031036 003720          $MXCNT: 2000.      ;; FIXES PS
7155 031040          $SWOBTBL:      ;; MAX. NUMBER OF ITERATIONS
7156 031040 003100          .WORD   TST1+2      ;; STARTING ADDRESS OF TEST 1
7157 031042 003224          .WORD   TST2+2      ;; STARTING ADDRESS OF TEST 2
7158 031044 003320          .WORD   TST3+2      ;; STARTING ADDRESS OF TEST 3
7159 031046 004352          .WORD   TST4+2      ;; STARTING ADDRESS OF TEST 4
7160 031050 004454          .WORD   TST5+2      ;; STARTING ADDRESS OF TEST 5
7161 031052 004632          .WORD   TST6+2      ;; STARTING ADDRESS OF TEST 6
7162 031054 004760          .WORD   TST7+2      ;; STARTING ADDRESS OF TEST 7
7163 031056 005064          .WORD   TST10+2     ;; STARTING ADDRESS OF TEST 10
7164 031060 006010          .WORD   TST11+2     ;; STARTING ADDRESS OF TEST 11
7165 031062 006126          .WORD   TST12+2     ;; STARTING ADDRESS OF TEST 12
7166 031064 006236          .WORD   TST13+2     ;; STARTING ADDRESS OF TEST 13
7167 031066 006500          .WORD   TST14+2     ;; STARTING ADDRESS OF TEST 14
7168 031070 006632          .WORD   TST15+2     ;; STARTING ADDRESS OF TEST 15
7169 031072 007036          .WORD   TST16+2     ;; STARTING ADDRESS OF TEST 16
7170 031074 007226          .WORD   TST17+2     ;; STARTING ADDRESS OF TEST 17
7171 031076 007350          .WORD   TST20+2     ;; STARTING ADDRESS OF TEST 20
7172 031100 010136          .WORD   TST21+2     ;; STARTING ADDRESS OF TEST 21
7173 031102 010526          .WORD   TST22+2     ;; STARTING ADDRESS OF TEST 22
7174 031104 010732          .WORD   TST23+2     ;; STARTING ADDRESS OF TEST 23
7175 031106 011142          .WORD   TST24+2     ;; STARTING ADDRESS OF TEST 24
7176 031110 011260          .WORD   TST25+2     ;; STARTING ADDRESS OF TEST 25
7177 031112 011404          .WORD   TST26+2     ;; STARTING ADDRESS OF TEST 26
7178 031114 011554          .WORD   TST27+2     ;; STARTING ADDRESS OF TEST 27
7179 031116 011700          .WORD   TST30+2     ;; STARTING ADDRESS OF TEST 30
7180 031120 012162          .WORD   TST31+2     ;; STARTING ADDRESS OF TEST 31
7181 031122 012416          .WORD   TST32+2     ;; STARTING ADDRESS OF TEST 32
7182 031124 012604          .WORD   TST33+2     ;; STARTING ADDRESS OF TEST 33
7183 031126 013034          .WORD   TST34+2     ;; STARTING ADDRESS OF TEST 34
7184 031130 013222          .WORD   TST35+2     ;; STARTING ADDRESS OF TEST 35

```

```

7185 031132 013446 .WORD TST36+2 ;:STARTING ADDRESS OF TEST 36
7186 031134 013634 .WORD TST37+2 ;:STARTING ADDRESS OF TEST 37
7187 031136 014026 .WORD TST40+2 ;:STARTING ADDRESS OF TEST 40
7188 031140 014222 .WORD TST41+2 ;:STARTING ADDRESS OF TEST 41
7189 031142 014274 .WORD TST42+2 ;:STARTING ADDRESS OF TEST 42
7190 031144 014346 .WORD TST43+2 ;:STARTING ADDRESS OF TEST 43
7191 031146 014650 .WORD TST44+2 ;:STARTING ADDRESS OF TEST 44
7192 031150 015054 .WORD TST45+2 ;:STARTING ADDRESS OF TEST 45
7193 031152 015260 .WORD TST46+2 ;:STARTING ADDRESS OF TEST 46
7194 031154 015476 .WORD TST47+2 ;:STARTING ADDRESS OF TEST 47
7195 031156 015702 .WORD TST50+2 ;:STARTING ADDRESS OF TEST 50
7196 031160 016106 .WORD TST51+2 ;:STARTING ADDRESS OF TEST 51
7197 031162 016232 .WORD TST52+2 ;:STARTING ADDRESS OF TEST 52
7198 031164 016356 .WORD TST53+2 ;:STARTING ADDRESS OF TEST 53
7199 031166 016502 .WORD TST54+2 ;:STARTING ADDRESS OF TEST 54
7200 031170 016626 .WORD TST55+2 ;:STARTING ADDRESS OF TEST 55
7201 031172 016752 .WORD TST56+2 ;:STARTING ADDRESS OF TEST 56
7202 031174 017016 .WORD TST57+2 ;:STARTING ADDRESS OF TEST 57
7203 031176 017070 .WORD TST60+2 ;:STARTING ADDRESS OF TEST 60
7204 031200 017340 .WORD TST61+2 ;:STARTING ADDRESS OF TEST 61
7205 031202 017514 .WORD TST62+2 ;:STARTING ADDRESS OF TEST 62
7206 031204 017744 .WORD TST63+2 ;:STARTING ADDRESS OF TEST 63
7207 031206 020124 .WORD TST64+2 ;:STARTING ADDRESS OF TEST 64
7208 031210 020622 .WORD TST65+2 ;:STARTING ADDRESS OF TEST 65
7209 031212 021026 .WORD TST66+2 ;:STARTING ADDRESS OF TEST 66
7210 031214 021132 .WORD TST67+2 ;:STARTING ADDRESS OF TEST 67
7211 031216 021552 .WORD TST70+2 ;:STARTING ADDRESS OF TEST 70
7212 031220 022172 .WORD TST71+2 ;:STARTING ADDRESS OF TEST 71
7213 031222 022302 .WORD TST72+2 ;:STARTING ADDRESS OF TEST 72
7214 031224 023134 .WORD TST73+2 ;:STARTING ADDRESS OF TEST 73
7215 031226 023612 .WORD TST74+2 ;:STARTING ADDRESS OF TEST 74
7216 031230 024026 .WORD TST75+2 ;:STARTING ADDRESS OF TEST 75
7217 031232 024274 .WORD TST76+2 ;:STARTING ADDRESS OF TEST 76
7218 031234 024354 .WORD TST77+2 ;:STARTING ADDRESS OF TEST 77
7219 031236 024510 .WORD TST100+2 ;:STARTING ADDRESS OF TEST 100
7220 031240 024734 .WORD TST101+2 ;:STARTING ADDRESS OF TEST 101
7221 031242 025062 .WORD TST102+2 ;:STARTING ADDRESS OF TEST 102
7222 031244 025406 .WORD TST103+2 ;:STARTING ADDRESS OF TEST 103
7223 031246 026132 .WORD TST104+2 ;:STARTING ADDRESS OF TEST 104
7224 031250 026272 .WORD TST105+2 ;:STARTING ADDRESS OF TEST 105
7225 031252 026442 .WORD TST106+2 ;:STARTING ADDRESS OF TEST 106
7226 031254 026612 .WORD TST107+2 ;:STARTING ADDRESS OF TEST 107
7227 031256 027310 .WORD TST110+2 ;:STARTING ADDRESS OF TEST 110
7228 031260 030066 .WORD SEOP+2 ;ADDRESS OF END OF PASS

```

.SBTTL APT COMMUNICATIONS ROUTINE

```

7229
7230
7231
7232 *****
7233 031262 112737 000001 031526 $ATY1: MOVB #1,$FFLG ;:TO REPORT FATAL ERROR
7234 031270 112737 000001 031524 $ATY3: MOVB #1,$MFLG ;:TO TYPE A MESSAGE
7235 031276 000403 BR $ATYC
7236 031300 112737 000001 031526 $ATY4: MOVB #1,$FFLG ;:TO ONLY REPORT FATAL ERROR
7237 031306 $ATYC:
7238 031306 010046 MOV R0,-(SP) ;:PUSH R0 ON STACK
7239 031310 010146 MOV R1,-(SP) ;:PUSH R1 ON STACK
7240 031312 105737 031524 TSTB $MFLG ;:SHOULD TYPE A MESSAGE?

```

```

7241 031316 001450          BEQ      5$          ;; IF NOT: BR
7242 031320 122737 000001 001316  CMPB   #APTENV,$ENV  ;; OPERATING UNDER APT?
7243 031326 001031          BNE     3$          ;; IF NOT: BR
7244 031330 132737 000100 001317  BITB   #APTPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
7245 031236 001425          BEQ     3$          ;; IF NOT: BR
7246 031340 017600 000004          MOV    24(SP),RO  ;; GET MESSAGE ADDR.
7247 031344 062766 000002 000004  ADD    #2,4(SP)  ;; BUMP RETURN ADDR.
7248 031352 005737 001276          TST    $MSGTYPE  ;; SEE IF DONE W/ LAST XMISSION?
7249 031356 001375          BNE     1$          ;; IF NOT: WAIT
7250 031360 010037 001312          MOV    RO,$MSGAD ;; PUT ADDR IN MAILBOX
7251 031364 105720          TSTB   (RO)+     ;; FIND END OF MESSAGE
7252 031366 001376          BNE     2$          ;;
7253 031370 163700 001312          SUB    $MSGAD,RO  ;; SUB START OF MESSAGE
7254 031374 006200          ASR    RO         ;; GET MESSAGE LNTH IN WORDS
7255 031376 010037 001314          MOV    RO,$MSGLGT ;; PUT LENGTH IN MAILBOX
7256 031402 012737 000004 001276  MOV    #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
7257 031410 000413          BR     5$          ;;
7258 031412 017637 000004 031436 3$: MOV    24(SP),4$  ;; PUT MSG ADDR IN JSR LINKAGE
7259 031420 062766 000002 000004  ADD    #2,4(SP)  ;; BUMP RETURN ADDRESS
7260 031426 013746 177776          MOV    177776,-(SP) ;; PUSH 177776 ON STACK
7261 031432 004737 041156          JSR    PC,$TYPE  ;; CALL TYPE MACRO
7262 031436 000000          .WORD 0
7263 031440          4$:
7264 031440 105737 031526          5$:
7265 031444 001416          10$: TSTB   $FFLG     ;; SHOULD REPORT FATAL ERROR?
7266 031446 005737 001316          BEQ    12$        ;; IF NOT: BR
7267 031452 001413          TST    $ENV      ;; RUNNING UNDER APT?
7268 031454 005737 001276          BEQ    12$        ;; IF NOT: BR
7269 031460 001375          TST    $MSGTYPE  ;; FINISHED LAST MESSAGE?
7270 031462 017637 000004 001300  BNE     11$       ;; IF NOT: WAIT
7271 031470 062766 000002 000004  MOV    24(SP),$FATAL ;; GET ERROR #
7272 031476 005237 001276          ADD    #2,4(SP)  ;; BUMP RETURN ADDR.
7273 031502 105037 031526          INC    $MSGTYPE  ;; TELL APT TO TAKE ERROR
7274 031506 105037 031525          CLRB   $FFLG     ;; CLEAR FATAL FLAG
7275 031512 105037 031524          CLRB   $LFLG     ;; CLEAR LOG FLAG
7276 031516 012601          CLRB   $MFLG     ;; CLEAR MESSAGE FLAG
7277 031520 012600          MOV    (SP)+,R1  ;; POP STACK INTO R1
7278 031522 000207          MOV    (SP)+,RO  ;; POP STACK INTO RO
7279 031524 000          RTS     PC       ;; RETURN
7280 031525 000          $MFLG: .BYTE 0  ;; MESSG. FLAG
7281 031526 000          $LFLG: .BYTE 0  ;; LOG FLAG
7282          000          $FFLG: .BYTE 0  ;; FATAL FLAG
7283          031530          .EVEN
7284          000200  APTSIZE=200
7285          000001  APTENV=001
7286          000100  APTPOOL=100
7287          000040  APTCSUP=040
7288          .SBTTL  ERROR HANDLER ROUTINE
7289          *****
7290          *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7291          *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7292          *AND GO TO TYPERR ON ERROR
7293          *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7294          *SW15=1  HALT ON ERROR
7295          *SW13=1  INHIBIT ERROR TYPEOUTS
7296          *SW10=1  BELL ON ERROR

```

```

7297
7298
7299
7300
7301 031530
7302 031530 104407
7303 031532 105237 001103
7304 031536 001775
7305 031540 013777 001102 147374
7306 031546 032777 002000 147364
7307 031554 001402
7308 031556 104401 001266
7309 031562 005237 001112
7310 031566 011637 001116
7311 031572 162737 000002 001116
7312 031600 117737 147312 001114
7313 031606 032777 020000 147324
7314 031614 001055
7315 031616 021627 001002
7316 031622 101046
7317
7318 031624 016637 000004 001116
7319 031632 162737 000002 001116
7320 031640 104401 031704
7321 031644 013746 001116
7322 031650 104402
7323 031652 104401 031712
7324 031656 162716 000004
7325 031662 011637 001116
7326 031666 013746 001116
7327 031672 104402
7328 031674 104401 001273
7329 031700 022626
7330 031702 000422
7331 031704 050200 036503 000040
7332 031712 020040 047125 054105
7333 031720 042520 052103 042105
7334 031726 052040 040522 020120
7335 031734 047524 000040
7336
7337 031740
7338 031740 004737 032052
7339 031744 104401 001273
7340 031750
7341 031750 122737 000001 001316
7342 031756 001007
7343 031760 113737 001114 031772
7344 031766 004737 031300
7345 031772 000
7346 031773 000
7347 031774 000777
7348 031776 005777 147136
7349 032002 100002
7350 032004 000000
7351 032006 104407
7352 032010 032777 001000 147122

```

```

;*SW09=1      LOOP ON ERROR
;*CALL
;*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER

$ERROR:
75:  CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
     INCB          ;;SET THE ERROR FLAG
     BEQ 7$        ;;DON'T LET THE FLAG GO TO ZERO
     MOV $STNM,$DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
     BIT #BIT10,$SWR ;;BELL ON ERROR?
     BEQ 1$        ;;NO - SKIP
     TYPE $BELL    ;;RING BELL
15:  INC $ERTTL    ;;COUNT THE NUMBER OF ERRORS
     MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
     SUB #2,$ERRPC
     MOVB $ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
     BIT #BIT13,$SWR ;;SKIP TYPEOUT IF SET
     BNE 20$       ;;SKIP TYPEOUTS
     CMP (SP),#1002 ;;IF RETURN PC LESS THAN 1002
     BHI 12$       ;;ERROR IS ILLEGAL TRAP
;;PROCESS UNEXPECTED TRAP OR INTERRUPT
     MOV 4(SP),$ERRPC ;;GET PC AT TIME OF FALSE TRAP
     SUB #2,$ERRPC    ;;ADJUST PC
     TYPE 10$        ;;TYPE HEADER
     MOV $ERRPC,-(SP) ;;SAVE $ERRPC FOR TYPEOUT
     TYPOC           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
     TYPE ,11$
     SUB #4,(SP)    ;;GET FALSE TRAP VECTOR ADDR
     MOV (SP),$ERRPC
     MOV $ERRPC,-(SP) ;;SAVE $ERRPC FOR TYPEOUT
     TYPOC           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
     TYPE $CRLF
     CMP (SP)+,(SP)+ ;;POP FALSE TRAP VECTOR PC&ADDR
     BR 20$
10$: .ASCIZ '<200>'PC= '
11$: .ASCIZ ' UNEXPECTED TRAP TO '

.EVEN
12$: JSR PC,TYPERR ;;GO TO USER ERROR ROUTINE
     TYPE $CRLF
20$: CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
     BNE 2$        ;;NO SKIP APT ERROR REPORT
     MOVB $ITEMB,21$ ;;SET ITEM NUMBER AS ERROR NUMBER
     JSR PC,$ATY4  ;;REPORT FATAL ERROR TO APT
21$: .BYTE 0
     .BYTE 0
22$: BR 22$       ;;APT ERROR LOOP
25$: TST $SWR     ;;HALT ON ERROR
     BPL          ;;SKIP IF CONTINUE
     HALT         ;;HALT ON ERROR!
35$: CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
     BIT #BIT09,$SWR ;;LOOP ON ERROR SWITCH SET?

```

```

7353 032016 001402          BEQ      4$          ;;BR IF NO
7354 032020 013716 001110    MOV      $LPERR,(SP)  ;;FUDGE RETURN FOR LOOPING
7355 032024 005737 001264    4$:    TST      $ESCAPE  ;;CHECK FOR AN ESCAPE ADDRESS
7356 032030 001402          BEQ      5$          ;;BR IF NONE
7357 032032 013716 001264    MOV      $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
7358 032036          5$:    CMP      #SENDAD,2#42 ;;ACT-11 AUTO-ACCEPT?
7359 032036 022737 030152 000042 BNE      6$          ;;BRANCH IF NO
7360 032044 001001          BNE      6$          ;;BRANCH IF NO
7361 032046 000000          HALT          ;;YES
7362 032050          6$:    RTI          ;;RETURN
7363 032050 000002          RTI          ;;RETURN
7364          ;;*****
7365          .SBTTL  TYPE ERROR ROUTINE
7366          ;*ENTRY JSR PC,TYPERR
7367          ;*RETURN RTS PC
7368          ;*
7369          ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
7370          ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
7371          ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
7372          ;*THE ERROR.
7373          ;;*****
7374 032052 104413          TYPERR: SAVREG
7375 032054 113737 001102 001302  MOVB     $STNM,$STNM  ;GET TEST NUMBER FOR REPORT
7376 032062 042737 177400 001302  BIC      #177400,$STNM ;CLEAR UNUSED BITS
7377 032070 113700 001114          MOVB     $ITEMB,R0    ;ENTER ERROR NUMBER
7378 032074 042700 177400          BIC      #177400,R0   ;CLEAR UNUSED BITS
7379 032100 005300          DEC     R0           ;FORM INDEX FOR ERROR TABLE
7380 032102 006300          ASL     R0
7381 032104 006300          ASL     R0
7382 032106 006300          ASL     R0
7383 032110 062700 001360 1$:    ADD     #ERRTB,R0     ;FORM ADDRESS OF ERROR ENTRY
7384 032114 012037 032130          MOV     (R0)+,2$     ;GET EM POINTER
7385 032120 001404          BEQ     3$           ;BRANCH IF THERE ISN'T ONE
7386 032122 104401 001273          TYPE   ,SCRLF       ;TYPE CARRIAGE RETURN LINE FEED
7387 032126 104401          TYPE   ;TYPE ERROR MESSAGE (EM)
7388 032130 000000          .WORD  0            ;EM POINTER GOES HERE
7389 032132 012037 032146 3$:    MOV     (R0)+,4$     ;GET DH POINTER
7390 032136 001404          BEQ     5$           ;BRANCH IF THERE ISN'T ONE
7391 032140 104401 001273          TYPE   ,SCRLF       ;TYPE CR-LF
7392 032144 104401          TYPE   ;TYPE DATA HEADER
7393 032146 000000          .WORD  0            ;DH POINTER GOES HERE
7394 032150 012001 5$:    MOV     (R0)+,R1     ;GET DT POINTER
7395 032152 001445          BEQ     20$          ;BRANCH IF THERE ARE NONE
7396 032154 005004          CLR     R4           ;SET INDENT SWITCH
7397 032156 012000          MOV     (R0)+,R0     ;GET DF POINTER
7398 032160 012002          MOV     (R0)+,R2     ;STORE NUMBER OF DH'S
7399 032162 104401 001273          TYPE   ,SCRLF
7400 032166 112003 10$:   MOVB     (R0)+,R3     ;GET & STORE NUMBER OF DATA WORDS
7401 032170 105720          TSTB   (R0)+        ;BUMP PAST FORMAT WORD
7402 032172 005703          TST     R3          ;TEST IF ANY DATA FOR THIS HEADER
7403 032174 001416          BEQ     14$          ;NO - SKIP DATA PRINT
7404 032176 005704          TST     R4           ;CHECK FOR INDENT
7405 032200 001004          BNE     12$          ;YES, GO INDENT
7406 032202 013146 11$:   MOV     2(R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
7407 032204 104402          TYPOC ;TYPE IT
7408 032206 005303          DEC     R3          ;MORE DATA WORDS

```

```

7409 032210 001403          BEQ      13$          ;NO-BRANCH
7410 032212 104401 045534 12$:     TYPE      ,SPACE2 ;TYPE SEPARATORS
7411 032216 000771          BR        11$          ;LOOP
7412 032220 104401 001273 13$:     TYPE      ,$CRLF   ;TYPE <CR><LF>
7413 032224 005710          TST      (R0)         ;CHECK IF NEXT HEADER AVAILABLE
7414 032226 001401          BEQ      14$          ;NO, DO NOT CHANGE INDENT
7415 032230 005104          COM      R4           ;CHANGE INDENT
7416 032232 005302          DEC      R2           ;MORE DH'S?
7417 032234 003414          BLE      20$         ;NO-BRANCH
7418 032236 012037 032256 15$:     MOV      (R0)+,18$    ;GET NEXT DH POINTER
7419 032242 001751          BEQ      10$         ;IF 0 GET DATA
7420 032244 005704          TST      R4           ;INDENT?
7421 032246 001402          BEQ      17$         ;NO, BRANCH
7422 032250 104401 045534          TYPE      ,SPACE2   ;TYPE INDENT
7423 032254 104401          17$:     TYPE      ;TYPE DH
7424 032256 000000          18$:     .WORD     0           ;DH POINTER GOES HERE
7425 032260 104401 001273          TYPE      , $CRLF   ;
7426 032264 000740          BR        10$         ;GET DATA
7427 032266 104414          20$:     RESREG
7428 032270 005237 001632          INC      ERRCNT      ;INCREMENT THE ERROR COUNT
7429 032274 032777 010000 146636          BIT      #SW12, $SWR  ;CHECK IF SWITCH 12 SET
7430 032302 001421          BEQ      25$         ;NO, RETURN
7431 032304 022737 000024 001632          CMP      #20.,ERRCNT ;CHECK IF ERROR THRESHOLD EXCEEDED
7432 032312 103015          BHS      25$         ;NO, RETURN
7433 032314 104401 047771          TYPE      ,ABORT    ;TYPE "PROGRAM ABORTED BECAUSE
7434                                ;ERROR THRESHOLD EXCEEDED"
7435 032320 005737 000042          TST      42          ;CHECK IF CHAIN MODE
7436 032324 001407          BEQ      22$         ;NO, HALT PROCESSOR
7437 032326 012706 001100          MOV      #STACK,SP   ;INITIALIZE STACK
7438 032332 012737 000001 030112          MOV      #1,$EOPCT   ;FORCE END OF PROGRAM
7439 032340 000137 030064          JMP      $EOP
7440 032344 000000          22$:     HALT
7441 032346 000207          25$:     RTS      PC
7442                                .SBTTL  NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
7443                                ;*
7444                                ;* THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.
7445                                ;* THIS IS THE INDICATION TO THE ROUTINE CHECKING
7446                                ;* NON-EXISTANT MEMORY THAT AN INTERRUPT DID OCCUR.
7447 032350 005237 001662          NEXINT: INC      INTSET ;BUMP THE INTERRUPT COUNTER
7448 032354 000002          RTI
7449
7450                                .SBTTL  RK611 INTERRUPT HANDLER
7451                                ;*
7452                                ;* MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE.  ACTUAL
7453                                ;* PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
7454                                ;* PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
7455                                ;* INTERRUPT OCCURRED.
7456 032356 005237 001662          INTHLR: INC      INTSET ;BUMP THE INTERRUPT FLAG
7457 032362 000002          RTI                ;RETURN.
7458
7459                                .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
7460                                ;*
7461                                ;* MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE.  THE REPORT
7462                                ;* WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
7463 032364 032777 020000 146546          PERHLR: BIT      #BIT13,$SWR ;TEST IF INHIBIT REPORT
7464 032372 001003          BNE      1$          ;YES - SKIP
  
```

```

7465 032374 104401 050220          TYPE      ,EM3          ;TYPE PARITY ERROR MESSAGE
7466 032400 104402          TYPOC          ;AND PC VALUE
7467 032402 012737 000001 030112 1$: MOV      #1,SEOPCT      ;FORCE END OF PROGRAM
7468 032410 012706 001100          MOV      #STACK,SP    ;CLEAN OFF STACK
7469 032414 000137 030064          JMP      SEOP
7470
7471          .SBTTL  LINE CLOCK INTERRUPT HANDLER
7472          ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
7473          ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
7474
7475 032420 005237 001660          LCKHLR: INC      LCLKTK          ;INCREMENT CLOCK TICK COUNTER
7476 032424 042777 000200 147242          BIC      #BIT7,JKWLADD      ;CLEAR MONITOR BIT
7477 032432 000002          RTI
7478
7479          ;*****
7480          .SBTTL  OPTION PRESENT TEST AND SETUP
7481          ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
7482          ;*      LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED
7483          ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
7484          OPTTST: SAVREG
7485 032434 104413          MOV      #4,R1          ;SET POINTER TO NEM VECTOR ADDRESS
7486 032436 012701 000004          MOV      (R1)+,-(SP)      ;STORE VECTOR CONTENTS
7487 032442 012146          MOV      (R1),-(SP)
7488 032444 011146          MOV      #PR7,(R1)        ;SET PRIORITY
7489 032446 012711 000340          MOV      #NEXINT,-(R1)    ;SET VECTOR TO NEM TEST HANDLER
7490 032452 012741 032350          CLR      INTSET          ;CLEAR INTERRUPT COUNTER
7491 032456 005037 001662          MOV      #1,MMCSR1        ;LOAD CSR FOR BANK 2
7492 032462 012777 000001 147210          NOP
7493 032470 000240          NOP
7494 032472 000240          NOP
7495 032474 000240          TST      INTSET          ;CHECK IF INTERRUPT OCCURRED
7496 032476 005737 001662          ;BECAUSE OF NEM ON PARITY OPTION REFER.
7497 032502 001011          BNE      2$              ;YES - SKIP PARITY SETUP
7498 032504 052737 000200 001656          BIS      #PARBK0,OPTFLG    ;SET PARITY OPTION FLAG
7499 032512 013700 001704          MOV      MMVECA,R0        ;SET POINTER TO VECTOR
7500 032516 012720 032364          MOV      #PERHLR,(R0)+    ;INSERT HANDLER ADDRESS
7501 032522 012710 000340          MOV      #PR7,(R0)        ;INSERT PRIORITY
7502 032526 005037 001662          CLR      INTSET          ;CLEAR INTERRUPT FLAG
7503 032532 012777 000001 147142          MOV      #1,MMCSR2        ;LOAD CSR FOR BANK 1
7504 032540 000240          NOP
7505 032542 000240          NOP
7506 032544 000240          NOP
7507 032546 005737 001662          TST      INTSET          ;TEST IN INTERRUPT OCCURRED
7508 032552 001003          BNE      3$              ;YES - SKIP SETUP
7509 032554 052737 000100 001656          BIS      #PARBK1,OPTFLG    ;SET FLAG
7510 032562 005037 001662          CLR      INTSET          ;CLEAR INTERRUPT COUNTER
7511 032566 013700 001676          MOV      KWLVEC,R0        ;SET POINTER TO VECTOR
7512 032572 012720 032420          MOV      #LCKHLR,(R0)+    ;INSERT ADDRESS OF INTERRUPT HCLR
7513 032576 012710 000340          MOV      #PR7,(R0)        ;INSERT PRIORITY
7514 032602 012777 000100 147064          MOV      #BIT6,JKWLADD    ;LOAD KW11-L FOR INTERRUPT ENABLE
7515 032610 000240          NOP
7516 032612 000240          NOP
7517 032614 000240          NOP
7518 032616 005737 001662          TST      INTSET          ;TEST IN NEM ON KW11-P REFERENCE
7519 032622 001003          BNE      4$              ;THIS BRANCH WILL BYPASS SET UP OF
7520          ;CLOCK OPTION
    
```

```

7521 032624 052737 100000 001656      BIS      #LCLKPR,OPTFLG ;SET CLOCK PRESENT FLAG
7522 032632 012701 000006      MOV      #6,R1       ;RESTORE OLD VECTOR
7523 032636 005037 001662      CLR      INTSET      ;CLEAR INT FLAG
7524 032642 012611      MOV      (SP)+,(R1)
7525 032644 012641      MOV      (SP)+,-(R1)
7526 032646 104414      RESREG
7527 032650 000207      RTS      PC
7528
7529
7530
7531
7532
7533
7534
7535 032652 032777 001000 146260 SCOP1$: BIT      #SW9,ASWR     ;CHECK IF LOOP ON ERROR
7536 032660 001405      BEQ      S$          ;NO. CONTINUE
7537 032662 105737 001103      TSTB     SERFLG     ;CHECK IF ERROR OCCURRED
7538 032666 001402      BEQ      S$
7539 032670 013716 001110      MOV      SLPERR,(SP) ;LOAD ERROR RETURN
7540 032674 000002      S$: RTI             ;RETURN
7541
7542
7543
7544
7545
7546
7547 032676 005037 001660      CLKCAL: CLR      LCLKTK   ;CLEAR TICK COUNTER
7548 032702 032737 100000 001656      BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
7549 032710 001004      BNE      1$         ;YES - SKIP
7550 032712 012737 000020 001654      MOV      #16.,TIMCNT ;ELSE PRESET TIMCNT
7551 032720 000410      BR      2$         ;AND EXIT
7552 032722 005737 001662      1$: TST      INTSET   ;TEST IF INTERRUPT HAS OCCURRED
7553 032726 001005      BNE      2$         ;YES- ABORT CALIBRATION
7554 032730 005737 001660      TST      LCLKTK    ;WAIT FOR CLOCK TICK
7555 032734 001772      BEQ      1$         ;NOT YET - LOOP
7556 032736 005037 001660      CLR      LCLKTK    ;CLEAR TICK COUNT
7557 032742 000207      2$: RTS      PC      ;RETURN
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573 032744 010046      MYTIME: MOV      R0,-(SP) ;SAVE R0
7574 032746 013700 001652      MOV      MILCNT,R0 ;SET COUNT
7575 032752 005737 001662      1$: TST      INTSET   ;TEST IF INTERRUPT SET
7576 032756 001012      BNE      2$         ;YES - SKIP
  
```

\*\*\*\*\*

.SBTTL LOOP ON INTERNAL ERROR  
 ;\* THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER  
 ;\* IS EXPECTED TO SET SLPERR TO THE START OF THE SCOPE LOOP  
 ;\* TO BE EXECUTED ON ERROR.

.SBTTL LINE CLOCK CALIBRATE  
 ;\* WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS  
 ;\* TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS  
 ;\* THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE  
 ;\* LINE CLOCK SIMULATOR.

.SBTTL LINE CLOCK SIMULATION ROUTINE  
 ;\* THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO  
 ;\* DO THIS THE VALUE STORED IN MILCNT IS USED AS THE  
 ;\* BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT  
 ;\* AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE  
 ;\* TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE  
 ;\* LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT  
 ;\* IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK  
 ;\* TICK).  
 ;\*  
 ;\* THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND  
 ;\* AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.



```

7577 032760 005300          DEC      RO          ;DECREMENT COUNT TO ZERO
7578 032762 001373          BNE     1$
7579 032764 005337 001654  DEC     TIMCNT      ;DECREMENT TIMCNT
7580 032770 001005          BNE     2$          ;IF NOT ZERO - EXIT
7581 032772 005237 001660  INC     LCLKTK      ;ELSE BUMP TICK COUNTER
7582 032776 012737 000020 001654  MOV     #16, TIMCNT ;RESET TIME COUNT
7583 033004 012600          MOV     (SP)+, RO   ;RESTORE RO
7584 033006 000207          RTS      PC        ;RETURN
7585
7586          .SBTTL  WAIT FOR INTERRUPT ROUTINE
7587          ;*     THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL
7588          ;*     SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE
7589          ;*     WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR
7590          ;*     IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED
7591          ;*     (BUT NOT PRINTED IN THE ROUTINE) AND THEN RETURNS TO THE
7592          ;*     LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE
7593          ;*     RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL
7594          ;*     BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.
7595
7596 033010 104413          IWAT8S: SAVREG      ;ENTRY FOR 8 SECOND WAIT
7597 033012 012700 000764  MOV     #500, RO
7598 033016 000463          BR     WATSRT
7599 033020 104413          IWAT1M: SAVREG      ;ENTRY FOR 1 MIN WAIT
7600 033022 012700 007246  MOV     #3750, RO
7601 033026 000457          BR     WATSRT
7602 033030 104413          IWAT2S: SAVREG      ;ENTRY FOR 2 SECOND WAIT
7603 033032 012700 000200  MOV     #128, RO
7604 033036 000453          BR     WATSRT
7605 033040 104413          IWAT1S: SAVREG      ;ENTRY FOR 1 SECOND WAIT
7606 033042 012700 000077  MOV     #63, RO
7607 033046 000447          BR     WATSRT
7608 033050 104413          IWAT159: SAVREG     ;ENTRY FOR 160 MS WAIT
7609 033052 012700 000012  MOV     #10, RO
7610 033056 000443          BR     WATSRT
7611 033060 104413          IWAT144: SAVREG     ;ENTRY FOR 144 MS WAIT
7612 033062 012700 000011  MOV     #9, RO
7613 033066 000437          BR     WATSRT
7614 033070 104413          IWAT128: SAVREG     ;ENTRY FOR 128 MS WAIT
7615 033072 012700 000010  MOV     #8, RO
7616 033076 000433          BR     WATSRT
7617 033100 104413          IWAT112: SAVREG     ;ENTRY FOR 112 MS WAIT
7618 033102 012700 000007  MOV     #7, RO
7619 033106 000427          BR     WATSRT
7620 033110 104413          IWAT96:  SAVREG     ;ENTRY FOR 96 MS WAIT
7621 033112 012700 000006  MOV     #6, RO
7622 033116 000423          BR     WATSRT
7623 033120 104413          IWAT80:  SAVREG     ;ENTRY FOR 80 MS WAIT
7624 033122 012700 000005  MOV     #5, RO
7625 033126 000417          BR     WATSRT
7626 033130 104413          IWAT64:  SAVREG     ;ENTRY FOR 64 MS WAIT
7627 033132 012700 000004  MOV     #4, RO
7628 033136 000413          BR     WATSRT
7629 033140 104413          IWAT48:  SAVREG     ;ENTRY FOR 48 MS WAIT
7630 033142 012700 000003  MOV     #3, RO
7631 033146 000407          BR     WATSRT
7632 033150 104413          IWAT32:  SAVREG     ;ENTRY FOR 32 MS WAIT

```

```

7633 033152 012700 000002          MOV    #2,RO
7634 033156 000403          BR     WATSRT
7635 033160 104413          IWAT16: SAVREG          ;ENTRY FOR 16 MS WAIT
7636 033162 012700 000001          MOV    #1,RO
7637 033166 012746 000000          WATSRT: MOV    #PRO,-(SP) ;LOAD PRIORITY 0 ON STACK
7638 033172 012746 033200          MOV    #5$,-(SP) ;LOAD ADDRESS
7639 033176 000802          RTI
7640
7641 033200 012737 000020 001654 5$:    MOV    #16,,TIMCNT ;PRESET TIME COUNTER
7642 033206 004737 032676          JSR    PC,CLKCAL ;GO CALIBRATE THE CLOCK
7643 033212 005737 001662          1$:    TST    INTSET ;TEST IF INTERRUPT OCCURRED
7644 033216 001036          BNE    3$ ;YES - EXIT
7645 033220 032737 100000 001656          BIT    #LCLKPR,OPTFLG ;TEST IF KW11-L AVAILABLE
7646 033226 001002          BNE    2$ ;YES - SKIP
7647 033230 004737 032744          JSR    PC,MYTIME ;ELSE CALL SIMULATOR
7648 033234 023700 001660          2$:    CMP    LCLKTK,RO ;TEST IF ENOUGH TICKS COUNTED
7649 033240 103764          BLO    1$ ;NO - LOOP
7650 033242 104420          TGETRK ;ELSE GET '611 REGS
7651 033244 013701 001540          MOV    T.CS1,R1 ;PUT CS1 IN R1- STRIP ALL BUT
7652 033250 042701 177741          BIC    #177741,R1 ;COMMAND CODE; INDEX INTO TABLE
7653 033254 016137 045074 001372          MOV    CMNDLB(R1),DH2N ;AND SELECT HEADER TO IDENTIFY OPERATION
7654 033262 012737 050270 001370          MOV    #EM4,EM2N ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
7655 033270 013700 001302          MOV    $TESTN,RO ;GET NUMBER OF PRESENT TEST
7656 033274 006300          ASL    RO ;SHIFT FOR INDEX
7657 033276 016037 031040 001264          MOV    $$WO8TB(RO),$ESCAPE ;LOAD ESCAPE TO ABORT TESTS
7658 033304 162737 000002 001264          SUB    #2,$ESCAPE ;BUT GO TO NEXT SCOPE CALL
7659 033312 000402          BR     4$
7660 033314 062716 000002          3$:    ADD    #2,(SP) ;BUMP RETURN AROUND ERROR
7661 033320 104414          4$:    RESREG ;RESTORE REGS
7662 033322 000002          RTI ;RETURN
7663
7664          .SBTTL "L" REGISTER LOADING ROUTINE
7665          ;* THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
7666          ;* THE "L" REGISTERS L.CS1-L.DCYL. L.MRI IS NOT
7667          ;* LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
7668          ;* FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
7669          ;*
7670          ;* CALL: JSR R4,LRLOAD
7671          ;* COMMAND
7672          ;* WORD COUNT
7673          ;* BUS ADDRESS
7674          ;* .BYTE SECTOR ADDRESS
7675          ;* .BYTE TRACK ADDRESS
7676          ;* CYLINDER ADDRESS
7677
7678          LRLOAD:
7679          MOV    RO,-(SP) ;: PUSH RO ON STACK
7680          MOV    R1,-(SP) ;: PUSH R1 ON STACK
7681          MOV    #L.CS1,R1 ;: GET ADDRESS OF L REGS
7682          MOV    #4,RO ;: PRESET COUNT
7683          1$:    MOV    (R4)+,(R1)+ ;: MOVE FIRST FOUR WORDS INTO "L" REGS
7684          DEC    RO ;: CS1, WC, BA, DA
7685          BNE    1$
7686          MOV    DRVNUM,(R1)+ ;: LOAD DRIVE NUMBER
7687          TST    (R1)+ ;: BUMP PAST ASOF
7688          MOV    (R4)+,(R1) ;: LOAD DCYL

```

```

7689 033356 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
7690 033360 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
7691 033362 000204      RTS      R4
7692
7693      .SBTTL  LOAD RK611 FOR OPERATION
7694      ;*    THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
7695      ;*    THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
7696      ;*    OF THE REGISTER CONTENTS.  L.CS1 IS TRANSFERRED LAST AS IT
7697      ;*    SHOULD BE IF THE GO BIT IS SET.
7698 033364 005037 001662  LOADRK: CLR      INTSET      ;CLEAR INTERRUPT FLAG.
7699 033370 010046      MOV      R0,-(SP)      ;STORE REGISTER
7700 033372 010146      MOV      R1,-(SP)
7701 033374 012700 001602  MOV      #L.WC,R0      ;GET ADDRESS OF SETUP STORAGE WC
7702 033400 010201      MOV      R2,R1        ;GET BASE ADDRESS
7703 033402 062701 000002  ADD      #2,R1        ;PUT R1 PAST RKCS1
7704 033406 012021      MOV      (R0)+,(R1)+  ;LOAD WORD COUNT
7705 033410 012021      MOV      (R0)+,(R1)+  ;LOAD BUS ADDRESS
7706 033412 012021      MOV      (R0)+,(R1)+  ;LOAD DISK ADDRESS
7707 033414 012011      MOV      (R0)+,(R1)   ;LOAD CS2
7708 033416 062701 000006  ADD      #6,R1        ;BUMP R1 TO ASOF
7709 033422 012021      MOV      (R0)+,(R1)+  ;LOAD OFFSET
7710 033424 012021      MOV      (R0)+,(R1)+  ;LOAD CYLINDER
7711 033426 062701 000004  ADD      #4,R1        ;BUMP R1 TO MR1
7712 033432 011011      MOV      (R0),(R1)    ;LOAD MR1
7713 033434 013712 001600  MOV      L.CS1,(R2)   ;LOAD CS1
7714 033440 012601      MOV      (SP)+,R1     ;RESTORE REGISTER
7715 033442 012600      MOV      (SP)+,R0
7716 033444 000002      RTI                    ;RETURN
7717
7718      .SBTTL  STORE RK611 REGISTERS
7719      ;*    ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
7720      ;*    WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN
7721      ;*    THIS ROUTINE.
7722
7723 033446 010046  GETRK: MOV      R0,-(SP)      ;STORE REGISTERS TO BE USED
7724 033450 010146      MOV      R1,-(SP)
7725 033452 010346      MOV      R3,-(SP)
7726 033454 012700 001540  MOV      #T.CS1,R0    ;SET POINTER TO TEST TABLE
7727 033460 010201      MOV      R2,R1        ;SET POINTER TO RK611 BASE
7728 033462 012703 000012  MOV      #10,R3       ;SET COUNT FOR 1ST 10 REGS
7729 033466 012120 1S:  MOV      (R1)+,(R0)+  ;STORE RKCS1 THROUGH RKSPAR
7730 033470 005303      DEC      R3
7731 033472 001375      BNE      1S
7732 033474 062701 000002  ADD      #2,R1        ;BUMP POINTER PAST RKDB
7733 033500 005720      TST      (R0)+        ;BUMP POINTER PAST T.RKDB
7734 033502 012703 000004  MOV      #4,R3        ;SET COUNT FOR LAST 5 REGS
7735 033506 012120 2S:  MOV      (R1)+,(R0)+  ;STORE RKMRI THROUTH RKMRI
7736 033510 005303      DEC      R3
7737 033512 001375      BNE      2S
7738 033514 012603      MOV      (SP)+,R3     ;RESTORE REGISTERS
7739 033516 012601      MOV      (SP)+,R1
7740 033520 012600      MOV      (SP)+,R0
7741 033522 000002      RTI                    ;RETURN
7742
7743      .SBTTL  BIT COUNTER IN A WORD
7744      ;*    THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK
7745      ;*    BY THE CALLING ROUTINE.  THE NUMBER OF BITS FOUND IN THE WORD

```

```

7745 ;* ARE PASSED BACK ON THE STACK.
7746
7747 033524 016637 000002 001256 BITCNT: MOV 2(SP),STMP16 ;GET WORD WHOSE BITS ARE TO BE COUNTED
7748 033532 010346 MOV R3,-(SP) ;STORE R3
7749 033534 005037 001260 CLR STMP17 ;CLEAR STMP16 FOR COUNTING
7750 033540 012703 000021 MOV #17.,R3 ;SET A SHIFT COUNTER
7751 033544 000241 CLC ;CLEAR CARRY
7752 033546 006037 001256 1$: ROR STMP16 ;ROTATE WORD.
7753 033552 103407 BCS 3$ ;WAS BIT SHIFTED OUT A 1?
7754 033554 005303 2$: DEC R3 ;NO - DEC COUNT
7755 033556 001373 BNE 1$ ;LOOP IF NOT ZERO
7756 033560 012603 MOV (SP)+,R3 ;RESTORE R3
7757 033562 013766 001260 000002 MOV STMP17,2(SF) ;PUT COUNT OF BITS ON STACK
7758 033570 000204 RTS R4 ;RETURN
7759 033572 005237 001260 3$: INC STMP17 ;BUMP COUNT
7760 033576 000766 BR 2$ ;LOOP
7761
7762 .SBTTL MAINTENANCE CLOCK ROUTINE
7763 ;* THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
7764 ;* ADDRESS OFTER THE CALL. THE FIRST BYTE CANTAINS THE NUMBER
7765 ;* OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
7766 ;* CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
7767 ;* CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
7768
7769 033600 MLOCK:
7770 033600 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
7771 033602 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
7772 033604 112400 MOVB (R4)+,R0 ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
7773 033606 112401 MOVB (R4)+,R1 ;GET PARTIAL PHASE ADDRESS COUNT
7774
7775 033610 006300 ASL R0 ;MULTIPLY PHASE ADDRESS COUNT BY 4
7776 033612 006300 ASL R0
7777 033614 060100 ADD R1,R0 ;ADD IN PARTIALS
7778 033616 052762 000400 000026 1$: BIS #MCLK,RKMR1(R2) ;SET CLOCK
7779 033624 042762 000400 000026 BIC #MCLK,RKMR1(R2) ;CLEAR MCLK
7780 033632 005300 DEC R0 ;DECREMENT COUNT
7781 033634 001370 BNE 1$ ;LOOP IF NOT ZERO
7782 033636 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
7783 033640 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
7784 033642 000204 RTS R4
7785 .SBTTL READ AND SORT HEADERS
7786 ;* THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
7787 ;* THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
7788 ;* ASSCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
7789 ;* READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
7790 ;* EACH READ OF THE DATA BUFFER.
7791
7792 CALL: JSR R4,RDSTHD
7793 TCHKOP ;RETURN POINT IF CERR IN READ HDR
7794 ERROR 4 ;OR 5, 6, 7
7795 ERROR 13 ;RETURN IF DATA LATE IN DB UNLOAD
7796 ERROR 2 ;RETURN IF TO SLOW OR
7797 ;IF HDR 0 NOT FOUND
7798
7799 033644 104413 RDSTHD: SAVREG
7800 033646 032737 100000 001656 BIT #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
    
```

7801	033654	001402			BEQ	20\$		:NO - SKIP
7802	033656	005077	146012		CLR	AKWLADD	;RESET INTERRUPT	
7803	033662	012700	000026		MOV	#26,R0		:PRESET FOR 26 SECTOR FORMAT
7804	033666	032737	010000	001600	BIT	#CFMT,L.CS1		:IS 24 SECTOR MODE SET?
7805	033674	001402			BEQ	1\$		:NO - SKIP
7806	033676	012700	000024		MOV	#24,R0		:ELSE CHANGE TO 24 SECTOR MODE
7807	033702	012701	056446		MOV	#IBUFF,R1		:SET POINTER TO INPUT BUFFER
7808	033706	010005			MOV	R0,R5		:SAVE NUMBER OF SECTORS
7809	033710	010104			MOV	R1,R4		:SAVE IBUFF ADDRESS
7810	033712	010203			MOV	R2,R3		:SET UP POINTER TO RKDB
7811	033714	062703	000024		ADD	#RKDB,R3		
7812	033720	013762	001626	000010	MOV	DRVNUM,RKCS2(R2)		:LOAD DRIVE NUM
7813	033726	013762	001614	000020	MOV	L.DCYL,RKDCYL(R2)		:LOAD CYLINDER NUM
7814	033734	013762	001606	000006	MOV	L.DA,RKDA(R2)		:LOAD TRACK AND SECTOR
7815								
7816	033742	012737	000020	001654	MOV	#16,TIMCNT		:SET TIME COUNTER
7817	033750	005037	001662		CLR	INTSET		:CLEAR INTERRUPT FLAG
7818	033754	005037	001660		CLR	LCLKTK		:CLEAR TICK COUNTER
7819	033760	013762	001600	000000	MOV	L.CS1,RKCS1(R2)		:LOAD COMMAND
7820								
7821	033766	005737	001662		TST	INTSET		:TEST IF INT OCCURRED
7822	033772	001020			BNE	4\$		:YES - SKIP
7823	033774	004737	032744		JSR	PC,MYTIME		:WAIT 1 MS
7824	034000	005737	001660		TST	LCLKTK		:HAVE WE WAITED 16 MS?
7825	034004	001770			BEQ	3\$		:NO - LOOP ON WAIT
7826								
7827	034006	062766	000006	000006	ADD	#6,6(SP)		:SET RETURN FOR TO SLOW
7828	034014	104420			TGETRK			:GET RK REGS
7829	034016	012737	050270	001370	MOV	#EM4,EM2N		:LOAD MESSAGE "TO SLOW/NOT COMPLETE"
7830	034024	012737	045300	001372	MOV	#OPER24,DH2N		:LOAD COMMAND "READ HEADER" FOR REPORT
7831	034032	000466			BR	10\$		:SKIP
7832								
7833	034034	005762	000000		TST	RKCS1(R2)		:TEST FOR CONTROLLER ERROR
7834	034040	100474			BMI	11\$		:YES - SKIP
7835								
7836	034042	011324			MOV	(R3),(R4)+		:STORE HEADERS
7837	034044	011324			MOV	(R3),(R4)+		
7838	034046	011324			MOV	(R3),(R4)+		
7839								
7840	034050	005762	000010		TST	RKCS2(R2)		:TEST IF DATA LATE
7841	034054	100443			BMI	8\$		:YES - SKIP
7842								
7843	034056	005300			DEC	R0		:DEC SECTOR COUNT
7844	034060	001330			BNE	2\$		:IF NOT ZERO - LOOP
7845								
7846	034062	032737	100000	001656	BIT	#LCLKPR,OPTFLG		:TEST IF CLOCK PRESENT
7847	034070	001403			BEQ	5\$		:NO - SKIP
7848	034072	012777	000100	145574	MOV	#BIT6,AKWLADD		:SET INTERRUPT ENABLE
7849	034100	032761	000037	000002	BIT	#37,2(R1)		:HEADER AT TOP OF BUFF=HEAD 0?
7850	034106	001413			BEQ	6\$		:YES - SKIP
7851	034110	012124			MOV	(R1)+,(R4)+		:ELSE MOV THIS HEADER TO BOTTOM
7852	034112	012124			MOV	(R1)+,(R4)+		
7853	034114	012124			MOV	(R1)+,(R4)+		
7854								
7855	034116	005305			DEC	R5		:TEST FO INSURE HEAD 0 IS FOUND
7856	034120	001367			BNE	5\$		:IF ALL HEADERS NOT CHECKED - LOOP

```

7857 034122 012737 053020 001370      MOV      #EM56,EM2N      ;ELSE "HEADER 0 NCT FOUND" MESSAGE
7858 034130 005037 001372      CLR      DH2N
7859 034134 000421      BR       9$             ;SKIP
7860
7861 034136 012700 056446      6$:     MOV      #IBUFF,RO      ;GET TOP OF IBUFF
7862 034142 012120      7$:     MOV      (R1)+,(RO)+      ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
7863 034144 012120      MOV      (R1)+,(RO)+
7864 034146 012120      MOV      (R1)+,(RO)+
7865 034150 020004      CMP     RO,R4           ;ALL HEADERS MOVED?
7866 034152 001373      BNE     7$             ;NO - LOOP
7867
7868 034154 062766 000010 000006      ADD     #10,6(SP)      ;SET UP FOR GOOD RETURN
7869 034162 000423      BR       11$
7870
7871 034164 012737 051303 001500 8$:     MOV      #EMDLT,EM13N     ;"DATA LATE SET RESULT OF READ DB"
7872 034172 012737 053431 055544      MOV      #EMRDB,DF011A
7873 034200 062766 000004 000006 9$:     ADD     #4,6(SP)        ;SET ERROR RETURN
7874 034206 104420      TGETRK      ;GET RK REGS
7875 034210 013700 001302 10$:    MOV      $TESTN,RO      ;GET TEST NUMBER
7876 034214 006300      ASL     RO             ;SHIFT FOR INDEX
7877 034216 016037 031040 001264      MOV      $$W08TB(RO),$ESCAPE ;SET ESCAPE
7878 034224 162737 000002 001264      SUB     #2,$ESCAPE     ;TO NEXT SCOPE CALL
7879
7880 034232 104414 11$:    RESREG
7881 034234 000204      RTS     R4
7882
7883      .SBTTL  GET DRIVE STATUS
7884      ;*     THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
7885      ;*     THROUGH $REG17.  THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
7886      ;*     THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
7887      ;*     IN THE REGISTERS.
7888      ;*
7889      ;*CALL: JSR     R4,GETDRS
7890      ;*     BR      ERROR PROCESSING      ERROR RETURN
7891      ;*     BR      NO ERROR PROCESSING   GOOD RETURN
7892
7893 034236 104413      GETDRS: SAVREG
7894 034240 012762 100000 000000      MOV     #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
7895 034246 012700 001202      MOV     #$REG10,RO      ;PRESET ALL STATUS STORAGE TO
7896 034252 012701 000010      MOV     #8,R1           ;ALL ONES
7897 034256 012720 177777      1$:    MOV     #177777,(RO)+
7898 034262 005301      DEC     R1
7899 034264 001374      BNE     1$
7900 034266 012700 001206      MOV     #$REG12,RO      ;SET POINTER TO REG12 FOR A01 & B01
7901 034272 012701 000001      MOV     #1,R1           ;PRESET FOR PAIR ONE.
7902 034276 005037 001230      CLR     $TMP3           ;CLEAR ERROR SWITCH
7903 034302 013762 001610 000010 2$:    MOV     L,CS2,RKCS2(R2) ;LOAD DRIVE #
7904 034310 010162 000026      MOV     R1,RKMR1(R2)    ;LOAD MR#
7905 034314 012762 000001 000000      MOV     #BIT0,RKCS1(R2) ;DO SELECT
7906 034322 012703 000050      MOV     #40.,R3        ;WAIT FOR A FEW MICRO RECORDS TO
7907 034326 005303      3$:    DEC     R3             ;BIT SELECT FINISH
7908 034330 001376      BNE     3$
7909 034332 032762 100000 000000      BIT     #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
7910 034340 001415      BEQ     4$             ;NO - SKIP
7911 034342 032762 024000 000000      BIT     #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
7912 034350 001004      BNE     8$             ;YES - SKIP

```

```

7913 034352 032762 037400 000010      BIT      #37400,RKCS2(R2) ;TEST FOR ERRORS:
7914                                     ; NED!UPE!MDS!UFE!NEM!PGE
7915 034360 001405                                     BEQ      4$          ;NO - SKIP
7916 034362 012737 000001 001230 8$:    MOV      #1,$TMP3    ;SET ERROR FLAG
7917 034370 022020                                     CMP      (R0)+,(R0)+ ;BUMP TO LET THAT PAIR STAY ALL 1'S.
7918 034372 000404                                     BR       5$          ;SKIP
7919 034374 016220 000034 4$:          MOV      RKMR2(R2),(R0)+ ;STORE A WORD
7920 034400 016220 000036             MOV      RKMR3(R2),(R0)+ ;STORE B WORD
7921 034404 012762 100000 000000 5$:    MOV      #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERROR IN CONTROLLER
7922 034412 005701                                     TST     R1          ;IS R1 A 0 (LAST TRANSFER, PAIR 0)
7923 034414 001410                                     BEQ     6$          ;YES - SKIP
7924 034416 005201                                     INC     R1          ;ELSE BUMP TO NEXT PAIR
7925 034420 022701 000004             CMP      #4,R1      ;PAIR 3 JUST STORED?
7926 034424 001326                                     BNE     2$          ;NO - SKIP
7927 034426 005001                                     CLR     R1          ;ELSE SET TO PAIR 0
7928 034430 012700 001202             MOV      #$REG10,R0 ;PRESET POINTER FOR PAIR 0
7929 034434 000722                                     BR      2$          ;GO GET THEM
7930 034436 104414 6$:          RESREG                                     ;EXIT HERE
7931 034440 005737 001230             TST     $TMP3      ;ANY ERROR IN STATUS GETTING
7932 034444 001001                                     BNE     7$          ;YES - SKIP
7933 034446 005724                                     TST     (R4)+      ;ELSE BUMP PART ERROR
7934 034450 000204 7$:          RTS      R4          ;RETURN
7935
7936                                     .SBTTL  SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
7937                                     ;*      THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
7938                                     ;*      COMMAND. CERR AND DI ARE MONITORED FOR A SHORT
7939                                     ;*      PEIROD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
7940                                     ;*
7941                                     ;*      IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
7942                                     ;*
7943                                     ;*      IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
7944                                     ;*      ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
7945                                     ;*      RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
7946                                     ;*      SKIPS OVER THE ERROR CALL AND TEST ABORT.
7947                                     ;*
7948                                     ;*      THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
7949                                     ;*      AN ERROR MESSAGE AND BRANCH TO END OF TEST. THIS
7950                                     ;*      IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
7951                                     ;*      THE TEST.
7952
7953                                     SSINIT:
7954 034452 010046             MOV      R0,-(SP)    ;: PUSH R0 ON STACK
7955 034454 010146             MOV      R1,-(SP)    ;: PUSH R1 ON STACK
7956 034456 012701 000007             MOV      #7,R1      ;: SET CLEAR COUNT
7957 034462 012700 001600             MOV      #L,CS1,R0  ;: GET ADDRESS OF "L" REGS
7958 034466 012720 000100             MOV      #100,(R0)+ ;: PRESET CS1
7959 034472 005020 7$:          CLR      (R0)+      ;: CLEAR THE NEXT
7960 034474 005301             DEC     R1          ;: COUNT 0?
7961 034476 001375             BNE     7$          ;: NO - LOOP
7962 034500 012762 000040 000010             MOV      #SCLR,RKCS2(R2) ;: CLEAR SUBSYSTEM
7963 034506 012737 000012 001222             MOV      #10,$TMP0  ;: SET A COUNTER
7964 034514 016237 000000 001540 1$:    MOV      RKCS1(R2),T.CS1 ;: GET CS1
7965 034522 032737 140000 001540             BIT     #CERR!DI,T.CS1 ;: TEST IF ERROR OR DI SET
7966 034530 001433             BEQ     2$          ;: NO - SKIP TO READY TEST
7967 034532 005337 001222             DEC     $TMP0      ;: ELSE DECREMENT COUNTER
7968 034536 001366             BNE     1$          ;: AND LOOP

```

```

7969 034540 032737 100000 001540 BIT #CERR,T.CS1 ;TEST - IS IT CERR STILL SET
7970 034546 001404 BEQ 3$ ;NO - SKIP TO DI MESSAGE
7971 034550 012737 050363 001400 MOV #EM5,EM3N ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
7972 034556 000403 BR 4$
7973 034560 012737 050427 001400 3$: MOV #EM6,EM3N ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
7974 034566 104420 4$: TGETRK
7975 034570 013700 001302 MOV $TESTN,RO ;GET PRESENT TEST NUMBER
7976 034574 006300 ASL RO ;SHIFT FOR INDEX
7977 034576 016037 031040 001264 MOV $$SWOBTBL(RO), $ESCAPE ;LOAD ESCAPE TO ABORT TEST
7978 034604 162737 000002 001264 SUB #2,$ESCAPE ;SET TO NEXT SCOPE CALL
7979 034612 012601 MOV (SP)+,R1 ;POP STACK INTO R1
7980 034614 012600 MOV (SP)+,RO ;POP STACK INTO RO
7981 034616 000414 BR 6$ ;SKIP TO EXIT
7982 034620 032737 000200 001540 2$: BIT #RDY,T.CS1 ;TEST READY SET
7983 034626 001004 BNE 5$ ;YES - GOOD EXIT
7984 034630 012737 050506 001400 MOV #EM7,EM3N ;MESSAGE (SUBSYS CLR NOT SET READY)
7985 034636 000753 BR 4$
7986 034640 012601 5$: MOV (SP)+,R1 ;RESTORE REGS
7987 034642 012600 MOV (SP)+,RO
7988 034644 062716 000002 ADD #2,(SP) ;GOOD RETURN
7989 034650 000002 6$: RTI
7990
7991 .SBTTL WORD COUNT AT END OF OPERATION CHECK
7992 ;* THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
7993 ;* THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF UNEQUAL, THE
7994 ;* ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
7995 ;*
7996 ;*CALL: JSR R4,CHKWC
7997 ;* .WORD ;EXPECTED WC VALUE
7998
7999 034652 012437 044720 CHKWC: MOV (R4)+,EXPWC ;STORE EXPECTED VALUE
8000
8001 034656 023737 044720 001542 CMP EXPWC,T.WC ;COMPARE
8002 034664 001406 BEQ 1$ ;EQUAL - SKIP
8003 034666 052737 000001 044756 BIS #WCERR,GRP4ER ;SET ERROR FLAG
8004 034674 013737 001542 044734 MOV T.WC,REALWC ;STORE REAL WORD COUNT
8005 034702 000204 1$: RTS R4 ;RETURN.
8006
8007 .SBTTL BUS ADDRESS AT END OF OPERATION CHECK
8008 ;* THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
8009 ;* A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
8010 ;* INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.
8011 ;* IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA
8012 ;* AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)
8013 ;*
8014 ;* IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
8015 ;* ADDRESS IS THE STARTING BUS ADDRESS.
8016 ;*CALL: JSR R4,CHKBA
8017
8018 034704 010046 CHKBA: MOV RO,-(SP)
8019 034706 010146 MOV R1,-(SP)
8020 034710 010346 MOV R3,-(SP)
8021 034712 032737 000020 001610 BIT #BAI,L.CS2 ;TEST IF BAI SET
8022 034720 001404 BEQ 4$ ;NO - SKIP
8023 034722 013737 001604 044724 MOV L.BA,EXPBA ;STORE EXPECTED BA (SAME AS STARTING BA)
8024 034730 000441 BR 3$

```



```

8025 034732 013700 001602 45: MOV L.WC,R0 ;GET INITIAL WORD COUNT
8026 034736 005400 NEG R0
8027 034740 113703 001601 MOVBL L.CS1+1,R3 ;GET BA16 & BA17
8028 034744 042703 111114 BIC #111114,R3 ;CLEAR UNWANTED BITS
8029
8030 034750 005700 TST R0 ;TEST IF INITIAL WORD COUNT 0
8031 034752 001003 BNE 65 ;NO - SKIP
8032 034754 062703 000002 ADD #2,R3 ;ADD 2 TO BA16,17 (65K WORD XFER)
8033 034760 000407 BR 95 ;SKIP
8034 034762 005700 65: TST R0 ;TEST IF INITIAL WC BIT 15 SET
8035 034764 100001 BPL 55 ;NO - SKIP
8036 034766 005203 INC R3 ;BUMP BA16,17 (32K WORD XFER)
8037 034770 006300 55: ASL R0 ;SHIFT WORD COUNT TO MAKE MEM ADD CNT
8038 034772 063700 001604 ADD L.BA,R0 ;ADD IN START BUFFER ADD
8039 034776 005503 ADC R3 ;IF CARRY - ADD INTO BA16,17
8040 035000 013701 001542 95: MOV T.WC,R1 ;GET END OF OPERATION WORD COUNT
8041 035004 001411 BEQ 15 ;BRANCH IF ZERO
8042 035006 005401 NEG R1
8043 035010 005701 TST R1 ;TEST END OPERATION WC BIT 15 SET
8044 035012 100001 BPL 75 ;NO - SKIP
8045 035014 005303 DEC R3 ;DEC BA 16,17 (32K WC LEFT)
8046 035016 006301 75: ASL R1 ;SHIFT WC TO MAKE MEM ADD CNT
8047 035020 160100 SUB R1,R0 ;SUB FROM COMPUTED BUS ADDRESS
8048 035022 005603 SBC R3 ;SUB CARRY FROM BA16,17
8049 035024 010337 044722 MOV R3,EXPUBA ;STORE EXPECTED UPPER BA BITS
8050 035030 010037 044724 15: MOV R0,EXPBA
8051 035034 020037 001544 35: CMP R0,T.BA ;EQUAL TO COMPUTED?
8052 035040 001406 BEQ 25 ;YES - SKIP
8053 035042 052737 000004 044756 BIS #BAERR,GRP4ER ;ELSE SET BAERR FLAG
8054 035050 013737 001544 044740 MOV T.BA,REALBA ;STORE REAL BUS ADDRESS
8055 035056 113703 001541 25: MOVBL T.CS1+1,R3 ;GET REAL UPPER BA
8056 035062 042703 177774 BIC #177774,R3 ;CLEAR UNWANTED BITS
8057 035066 020337 044722 CMP R3,EXPUBA ;CHECK = EQUAL
8058 035072 001405 BEQ 85 ;YES - SKIP
8059 035074 052737 000002 044756 BIS #UBAERR,GRP4ER ;ELSE SET UBA ERROR
8060 035102 010337 044736 MOV R3,REALUB ;STORE REAL UPPER BA
8061 035106 012603 85: MOV (SP)+,R3
8062 035110 012601 MOV (SP)+,R1
8063 035112 012600 MOV (SP)+,R0
8064 035114 000204 RTS R4
    
```

```

8065
8066 .SBTTL CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION
8067 : * THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA
8068 : * ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE
8069 : * OPERATION. THE CONTENTS OF THE LOAD REGISTER STORAGE ARE
8070 : * COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE
8071 : * NECESSARY CALCULATION.
8072 : *
8073 : * ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES
8074 : * STORAGE EXPCYL, EXPTRK, EXPSEC. ALL 3 ARE CHECKED AND
8075 : * IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE
8076 : * ERROR FLAGS FIELD (GRP4ER) IS SET.
8077 : *
8078 : *CALL: JSR R4,CHKCTS
8079
8080 035116 104413 CHKCTS: SAVREG
    
```

8081	035120	013700	001602		MOV	L.WC,R0	:GET SPECIFIED WORD COUNT	
8082	035124	005400			NEG	R0	:NEGATE IT	
8083	035126	013701	001542		MOV	T.WC,R1	:GET END OF OPERATION WORD COUNT	
8084	035132	001401			BEG	10\$	:IF ZERO - SKIP	
8085	035134	005401			NEG	R1	:NEGATE IT	
8086	035136	160100		10\$:	SUB	R1,R0	:COMPUTE ACTUAL WORDS TRANSFERRED	
8087	035140	005001			CLR	R1	:CLEAR R1 FOR COUNTING	
8088				:	THE FOLLOWING CODE DETERMINES HOW MANY SECTORS OF DATA HAS BEEN			
8089				:	TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE			
8090				:	END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR			
8091				:	IS CALCULATED.			
8092	035142	022700	000400	1\$:	CMP	#400,R0		
8093	035146	003004			BGT	2\$		
8094	035150	005201			INC	R1		
8095	035152	162700	000400		SUB	#400,R0		
8096	035156	000771			BR	1\$		
8097	035160	005700		2\$:	TST	R0		
8098	035162	001401			BEG	3\$		
8099	035164	005201			INC	R1		
8100				:	AT THIS POINT R1 HAS A COUNT OF THE			
8101				:	NUMBER OF FULL SECTOR TRANSFER + 1 IF A			
8102				:	PARTIAL SECTOR WAS TRANSFERRED.			
8103	035166	012703	000026	3\$:	MOV	#26,R3		
8104	035172	032737	010000	001600	BIT	#CFMT,L.CS1		
8105	035200	001402			BEG	4\$		
8106	035202	012703	000024		MOV	#24,R3		
8107				:	R3 HAS BEEN SET UP WITH THE NUMBER			
8108				:	OF SECTORS IN A TRACK FOR THE FORMAT USED			
8109	035206	013737	001614	044726	4\$:	MOV	L.DCYL,EXPCYL	:GET STARTING VALUES FOR CYLINDER
8110	035214	113704	001607		MOVB	L.DT,R4	:TRACK	
8111	035220	042704	177400		BIC	#177400,R4		
8112	035224	113705	001606		MOVB	L.DS,R5	:SECTOR	
8113	035230	042705	177400		BIC	#177400,R5		
8114	035234	005301			DEC	R1	:ADJUST COUNT FOR ZERO DETECT	
8115	035236	005205		5\$:	INC	R5	:BUMP SECTOR COUNT	
8116	035240	020503			CMP	R5,R3	:DID THIS MAKE SECTOR COUNT > 1 TRACK?	
8117	035242	001010			BNE	6\$	:NO - SKIP	
8118	035244	005005			CLR	R5	:ELSE CLEAR SECTOR COUNT	
8119	035246	005204			INC	R4	:BUMP TRACK COUNT	
8120	035250	022704	000003		CMP	#3,R4	:DID THIS MAKE TRK COUNT > 1 CYLINDER?	
8121	035254	001003			BNE	6\$	:NO - SKIP	
8122	035256	005004			CLR	R4	:ELSE CLEAR TRACK COUNT	
8123	035260	005237	044726		INC	EXPCYL	:BUMP CYLINDER COUNT	
8124	035264	005301		6\$:	DEC	R1	:DEC COUNT	
8125	035266	001363			BNE	5\$	:IF ZERO - EXIT	
8126	035270	010437	044732		MOV	R4,EXPTRK	:STORE EXPECTED TRACK	
8127	035274	010537	044730		MOV	R5,EXPSEC	:STORE EXPECTED SECTOR (CYL ALREADY SLOW)	
8128	035300	023737	001560	044726	CMP	T.DCYL,EXPCYL	:TEST IF CYLINDER OK	
8129	035306	001403			BEG	7\$	:YES - SKIP	
8130	035310	052737	000010	044756	BIS	#CYLERR,GRP4ER	:NO - SET ERROR FLAG	
8131	035316	120437	001547	7\$:	CMPB	R4,T.DA+1	:TEST TRACK OK	
8132	035322	001403			BEG	8\$	:YES - SKIP	
8133	035324	052737	000020	044756	BIS	#TRKERR,GRP4ER	:NO - SET ERROR FLAG	
8134	035332	120537	001546	8\$:	CMPB	R5,T.DA	:TEST SECTOR COUNT OK	
8135	035336	001403			BEG	9\$	:YES - SKIP	
8136	035340	052737	000040	044756	BIS	#SECERR,GRP4ER	:USE SET ERROR FLAG	

```

8137 035346 012700 044734          9$:  MOV    #REALWC,RO
8138 035352 013720 001542          MOV    T.WC,(RO)+      ;STORE REAL WORD COUNT
8139 035356 013720 001544          MOV    T.BA,(RO)+      ;STORE REAL BUS ADDRESS
8140 035362 013720 001560          MOV    T.DCYL,(RO)+    ;STORE REAL CYLINDER ADDRESS
8141 035366 113710 001547          MOVVB  T.DA+1,(RO)     ;STORE REAL TRACK ADDRESS
8142 035372 005720                      TST    (RO)+
8143 035374 113710 001546          MOVVB  T.DA,(RO)       ;STORE REAL SECTOR ADDRESS
8144 035400 104414                      RESREG
8145 035402 000204                      RTS     R4

8146
8147          .SBTTL  OPERATION CHECK ROUTINE
8148          ;*      THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME SOFTWARE
8149          ;*      ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:
8150          ;*      THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION
8151          ;*      FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH
8152          ;*      THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN
8153          ;*      ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED
8154          ;*      BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL
8155          ;*      BE THE PROPER ITEM FOR THE FORMAT REQUIRED BY THIS
8156          ;*      ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4,5,6,
8157          ;*      7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION
8158          ;*      OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT
8159          ;*      FORMAT.
8160
8161 035404 104413          CHKWE:  SAVREG
8162 035406 011600          MOV    (SP),RO        ;GET POINTER TO ERROR WORDS
8163 035410 012037 001242          MOV    (RO)+,$TMP10   ;STORE EXPECTED ERROR GROUP 1
8164 035414 012037 001244          MOV    (RO)+,$TMP11   ;STORE EXPECTED ERROR GROUP 2
8165 035420 012037 001246          MOV    (RO)+,$TMP12   ;STORE EXPECTED ERROR GROUP 3
8166 035424 010016          MOV    RO,(SP)        ;STORE RETURN
8167 035426 012737 177777 001250      MOV    #-1,$TMP13     ;SET FLAG - EXPECTED ERROR
8168 035434 000403          BR     CHKST

8169
8170 035436 104413          CHKOP:  SAVREG
8171 035440 005037 001250          CLR    $TMP13         ;RESET EXPECTED ERROR FLAG

8172
8173 035444 104420          CHKST:  TGETRK        ;GET 611 REGS IO TRAP
8174 035446 005037 044750          CLR    GRP1ER        ;CLEAR ERROR FLAGS
8175 035452 005037 044752          CLR    GRP2ER
8176 035456 005037 044754          CLR    GRP3ER
8177 035462 005037 044756          CLR    GRP4ER
8178 035466 005037 045072          CLR    GPSUMF        ;CLEAR SUMMARY FLAGS
8179 035472 032737 024000 001540      BIT    #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
8180 035500 001111          BNE   4$              ;YES - SKIP
8181 035502 032737 177400 001550      BIT    #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
8182 035510 001105          BNE   4$              ;YES - SKIP
8183 035512 032737 000070 001552      BIT    #DSERBIT,T.DS  ;TEST IF ERROR SET IN DS
8184 035520 001101          BNE   4$              ;YES - SKIP
8185 035522 005737 001554          TST    T.ER          ;TEST IF ERROR SET IN ER
8186 035526 001076          BNE   4$              ;YES - SKIP
8187 035530 032737 100000 001540      BIT    #CERR,T.CS1   ;COMBINED ERROR SET?
8188 035536 001405          BEQ   9$              ;NO - SKIP
8189 035540 052737 100000 044750      BIS    #CERNER,GRP1ER ;SET ERROR FLAG IN GROUP 1
8190 035546 000137 036276          JMP    25$           ;SKIP

8191
8192          ;      CODE TO CHECK WORD COUNT, BUFFER ADDRESS, CYLINDER, TRACK,
    
```

```

8193      ; AND SECTOR AT THE END OF THE OPERATION. THIS IS DONE ONLY
8194      ; IF CERR NOT SET BY THE OPERATION.
8195      ;
8196      ; ALL OF THE ABOVE CONDITIONS ARE CHECKED AND A BIT SET FOR
8197      ; EACH CHECK THAT FAILS. HOWEVER, ONLY ONE ERROR IS REPORTED.
8198      ; THE ORDER OF PRIORITY FOR REPORTING THE ERROR IS THE ORDER
8199      ; LISTED ABOVE.
8200
8201 035552 005737 001250 9$: TST $TMP13 ;TEST IF ERROR EXPECTED
8202 035556 001402 BEQ 62$ ;NO - SKIP
8203 035560 000137 036276 JMP 25$ ;YES - JUMP
8204 035564 013700 001540 62$: MOV T,CS1,RO ;GET CS1
8205 035570 042700 177741 BIC #177741,RO ;CHECK IF OPERATION IS READ DATA,
8206 035574 022700 000020 CMP #20,RO ;WRITE DATA, OR WRITE CHECK. IF
8207 035600 002445 BLT 3$ ;NOT, SKIP ALL CHECKING IN GROUP
8208 035602 022700 000030 CMP #30,RO ;FOUR
8209 035606 003042 BGT 3$
8210 035610 004437 034652 JSR R4,CHKWC ;CHECK WORD COUNT
8211 035614 000000 .WORD 0 ;EXPECTED WORD COUNT
8212 035616 004437 034704 JSR R4,CHKBA ;CHECK BUS ADDRESS
8213 035622 004437 035116 JSR R4,CHKCTS ;CHECK CYL, TRACK, & SECTOR
8214 035626 005737 044756 TST GRP4ER ;ANY GROUP 4 ERRORS?
8215 035632 001430 BEQ 3$ ;NO - SKIP
8216 035634 016037 045074 055524 MOV CMNDLB(RO),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
8217 035642 013700 044756 MOV GRP4ER,RO ;PUT GROUP 4 ERROR FLAG IN RO
8218 035646 005001 CLR R1 ;CLEAR R1 FOR INDEX COUNTER
8219 035650 006200 1$: ASR RO ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
8220 035652 103402 BCS 2$ ;BE REPORTED, REST ARE IGNORED
8221 035654 005720 TST (RO)+ ;WHEN AN ERROR BIT IS FOUND,
8222 035656 000774 BR 1$ ;GET THE ERROR MESSAGE ASSOCIATED
8223 035660 016037 044760 001450 2$: MOV GRP4MS(RO),EM10N ;WITH IT AND SET ERROR TABLE ITEM TO
8224 035666 016037 044720 001202 MOV EXPWC(RO),SREG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
8225 035674 016037 044734 001204 MOV REALWC(RO),SREG11 ;WITH EXPECTED & IS VALUES
8226 035702 104414 RESREG ;RESTORE REGISTER
8227 035704 012776 000010 000000 MOV #10,(SP) ;MAKE THE ERROR CALL POINT TO THE
8228 035712 000002 RTI ;RIGHT TABLE ENTRY & RETURN.
8229 035714 3$:
8230 035714 104414 RESREG
8231 035716 062716 000002 ADD #2,(SP) ;BUMP RETURN PAST ERROR
8232 035722 000002 RTI ;RETURN
8233
8234 ; THE FOLLOWING CODE BUILDS THE GROUP 1,2, & 3 ERROR WORDS.
8235
8236 035724 012700 044750 4$: MOV #GRP1ER,RO ;SET UP GENERAL REGISTER AS POINTER
8237 035730 012701 001540 MOV #T.CS1,R1 ;CS1
8238 035734 012703 001550 MOV #T.CS2,R3 ;CS2
8239 035740 012704 001552 MOV #T.DS,R4 ;DS
8240 035744 012705 001554 MOV #T.ER,R5 ;AND ER
8241
8242 035750 051510 BIS (R5),(RO) ;SET ALL BITS IN GRP1ER THAT
8243 ;CORRESPOND TO ERROR BITS IN RKER
8244 035752 042710 120701 BIC #ILF!ECH!BSE!HVRC!OPI!DCK,(RO) ; CLEAR ALL THAT DON'T BELONG GRP1
8245
8246 035756 032711 020000 BIT #SPAR,(R1) ;TEST IF SPAR SET
8247 035762 001402 BEQ 5$ ;NO - SKIP
8248 035764 052710 000001 BIS #SPARERR,(RO) ;SET SPAR ERROR FLAG
    
```

# F15

8249							
8250	035770	032714	000010	5\$:	BIT	#ACLO,(R4)	;TEST ACLO SET
8251	035774	001402			BEQ	6\$	;NO - SKIP
8252	035776	052710	000100		BIS	#ACLOERR,(RO)	;SET ACLO ERROR FLAG
8253							
8254	036002	032714	000020	6\$:	BIT	#SPDLSS,(R4)	;TEST SPEED LOSS SET
8255	036006	001402			BEQ	7\$	;NO - SKIP
8256	036010	052710	000200		BIS	#SPDERR,(RO)	;SET SPEED LOSS ERROR FLAG
8257							
8258	036014	032714	000040	7\$:	BIT	#DROT,(R4)	;TEST IF DROT SET
8259	036020	001402			BEQ	8\$	;NO - SKIP
8260	036022	052710	000400		BIS	#DROTERR,(RO)	;SET DROT ERROR FLAG
8261							
8262	036026	032711	100000	8\$:	BIT	#CERR,(R1)	;TEST CERR ITSELF SET
8263	036032	001002			BNE	10\$	;YES - SKIP
8264	036034	032710	020000		BIT	#NCERWE,(RO)	;SET NO CERR WITH ERROR FLAG
8265							
8266	036040	012700	044752	10\$:	MOV	#GRP2ER,RO	;SET POINTER TO GROUP 2 ERROR FLAGS
8267							
8268	036044	032715	000100		BIT	#ECH,(R5)	;TEST IF ECH SET
8269	036050	001402			BEQ	11\$	;NO - SKIP
8270	036052	052710	000001		BIS	#ECHERR,(RO)	;SET ECH FLAG
8271							
8272	036056	032715	100000	11\$:	BIT	#DCK,(R5)	;TEST DCK SET
8273	036062	001402			BEQ	12\$	;NO - SKIP
8274	036064	052710	000002		BIS	#DCKERR,(RO)	;SET DCK ERROR FLAG.
8275							
8276	036070	032713	040000	12\$:	BIT	#WCE,(R3)	;TEST WRITE CHECK ERROR
8277	036074	001402			BEQ	120\$	;NO - SKIP
8278	036076	052710	000004		BIS	#WCKERR,(RO)	;SET WCE BIT
8279	036102	032713	100000	120\$:	BIT	#DLT,(R3)	;TEST DATA LATE
8280	036106	001402			BEQ	13\$	;NO - SKIP
8281	036110	052710	000010		BIS	#DLTERR,(RO)	;SET DLT ERROR FLAG
8282							
8283	036114	032715	020000	13\$:	BIT	#OPI,(R5)	;TEST OPI SET
8284	036120	001402			BEQ	14\$	;NO - SKIP
8285	036122	052710	000020		BIS	#OPIERR,(RO)	;SET OPI ERROR FLAG
8286							
8287	036126	032715	000400	14\$:	BIT	#HVRC,(R5)	;TEST HVRC SET
8288	036132	001402			BEQ	16\$	;NO - SKIP
8289	036134	052710	000040		BIS	#HVRCERR,(RO)	;SET HVRC FLAG
8290							
8291	036140	032715	000200	16\$:	BIT	#BSE,(R5)	;TEST BSE ERROR FLAG
8292	036144	001402			BEQ	17\$	;NO - SKIP
8293	036146	052710	000100		BIS	#BSERR,(RO)	;SET BSE FLAG
8294							
8295	036152	012700	044754	17\$:	MOV	#GRP3ER,RO	;SET POINTER TO GROUP 3 FLAGS
8296							
8297	036156	032713	010000		BIT	#NED,(R3)	;TEST NED SET
8298	036162	001402			BEQ	18\$	;NO - SKIP
8299	036164	052710	000001		BIS	#NEDERR,(RO)	;SET NED FLAG
8300							
8301	036170	032711	004000	18\$:	BIT	#CTO,(R1)	;TEST CTO SET
8302	036174	001402			BEQ	19\$	;NO - SKIP
8303	036176	052710	000002		BIS	#CTOERR,(RO)	;SET CTO FLAG
8304							

8305	036202	032713	000400	19\$:	BIT	#UFE, (R3)	;TEST UFE SET
8306	036206	001402			BEQ	20\$	;NO - SKIP
8307	036210	052710	000004		BIS	#UFERR, (R0)	;SET UFE FLAG
8309							
8309	036214	032713	001000	20\$:	BIT	#MDS, (R3)	;TEST MDS SET
8310	036220	001402			BEQ	21\$	;NO - SKIP
8311	036222	052710	000010		BIS	#MDSERR, (R0)	;SET MDE FLAG
8312							
8313	036226	032713	002000	21\$:	BIT	#PGE, (R3)	;TEST PGE SET
8314	036232	001402			BEQ	22\$	;NO - SKIP
8315	036234	052710	000020		BIS	#PGERR, (R0)	;SET PGE FLAG
8316							
8317	036240	032713	004000	22\$:	BIT	#NEM, (R3)	;TEST NEM SET
8318	036244	001402			BEQ	23\$	;NO - SKIP
8319	036246	052710	000040		BIS	#NEMERR, (R0)	;SET NEM FLAG
8320							
8321	036252	032713	020000	23\$:	BIT	#UPE, (R3)	;TEST UPE SET
8322	036256	001402			BEQ	24\$	;NO - SKIP
8323	036260	052710	000100		BIS	#UPERR, (R0)	;SET UPE FLAG
8324							
8325	036264	032715	000001	24\$:	BIT	#ILF, (R5)	;TEST ILF SET
8326	036270	001402			BEQ	25\$	;NO - SKIP
8327	036272	052710	000200		BIS	#ILFERR, (R0)	;SET ILF FLAG.
8328							
8329				:			THE FOLLOWING CODE IS EXECUTED ONLY IF ERRORS WERE EXPECTED.
8330				:			THE FLAG IN \$TMP13 INDICATES IF
8331				:			AN ERROR WAS EXPECTED AND THE CONTENTS OF TMP10,
8332				:			TEMP11, & TEMP12 SPECIFY WHICH ERRORS. THESE ARE COMPARED AGAINST
8333				:			THE ERRORS FOUND AND STORED IN GRP1R, GRP2R, AND GRP3R.
8334				:			THE CONTENTS OF GRP1, 2, & 3 ARE MODIFIED TO INDICATE ERRORS THAT
8335				:			OCCURRED BUT WERE NOT EXPECTED. THE CONTENTS OF \$TMP10, 11,
8336				:			& 12 ARE MODIFIED TO INDICATE EXPECTED ERRORS THAT DID NOT
8337				:			OCCUR. BOTH CONDITIONS CAN EXIST AT THE SAME TIME AND MUST
8338				:			BE REPORTED.
8339				:			
8340	036276	005737	001250	25\$:	TST	\$TMP13	;CHECK IF AN ERROR WAS EXPECTED
8341	036302	001423			BEQ	110\$	;NO - SKIP
8342	036304	012704	044750		MOV	#GRP1R,R4	;GET ADDRESS OF ERROR
8343	036310	012705	001242		MOV	#\$TMP10,R5	;GET ADDRESS OF EXPECTED ERRORS
8344							
8345	036314	011500		26\$:	MOV	(R5),R0	;GET EXPECTED ERROR
8346	036316	011401			MOV	(R4),R1	;GET GROUP ERROR FLAGS
8347	036320	020001			CMP	R0,R1	;ARE THEY EQUAL?
8348	036322	001003			BNE	27\$	;NO - SKIP
8349	036324	005000			CLR	R0	;CLEAR EXPECTED ED
8350	036326	005001			CLR	R1	;CLEAR OCCURED
8351	036330	000403			BR	28\$	
8352							
8353	036332	010003		27\$:	MOV	R0,R3	;STORE EXPECTED ERRORS
8354	036334	040100			BIC	R1,R0	;RESET EXPECTED THAT OCCURRED
8355	036336	040301			BIC	R3,R1	;RESET OCCURRED THAT EXPECTED
8356	036340	010025		28\$:	MOV	R0,(R5)+	;STORE EXPECTED THAT DID NOT OCCUR
8357	036342	010124			MOV	R1,(R4)+	;STORE OCCURRED THAT WERE NOT EXPECTED
8358	036344	022705	001250		CMP	#\$TMP13,R5	;ALL GROUPS CHECKED.
8359	036350	001361			BNE	26\$	;NO - LOOP
8360							

8361  
 8362  
 8363  
 8364  
 8365  
 8366  
 8367  
 8368  
 8369  
 8370  
 8371  
 8372  
 8373  
 8374  
 8375  
 8376  
 8377  
 8378  
 8379  
 8380  
 8381  
 8382  
 8383  
 8384  
 8385  
 8386  
 8387 036352 005004  
 8388 036354 005005  
 8389 036356 012700 001224  
 8390 036362 012701 001226  
 8391 036366 012703 045072  
 8392 036372 012710 055504  
 8393 036376 012711 001442  
 8394 036402 013746 044754  
 8395 036406 004437 033524  
 8396 036412 005716  
 8397 036414 001403  
 8398 036416 061605  
 8399 036420 052713 000004  
 8400  
 8401 036424 005726  
 8402 036426 005737 001250  
 8403 036432 001412  
 8404 036434 013746 001246  
 8405 036440 004437 033524  
 8406 036444 005716  
 8407 036446 001403  
 8408 036450 052713 000040  
 8409 036454 061604  
 8410  
 8411 036456 005726  
 8412 036460 013746 044752  
 8413 036464 004437 033524  
 8414 036470 005716  
 8415 036472 001407  
 8416 036474 052713 000002

THE FOLLOWING CODE:

- A. DETERMINES WHICH FORMAT IS TO BE USED
- B. LOADS THE ADDRESSES OF THE ASCIZ TEXT INTO THE SELECTED ERROR TABLE ITEM AND FORMAT FIELD
- C. COUNTS THE NUMBER OF ERRORS THAT MUST BE REPORTED
- D. GETS DRIVE STATUS IF GROUP 1 ERROR.

THE DECISION OF WHICH ERROR IS TO BE USED IS BASED ON THE ERROR GROUP (OR GROUPS) THAT HAVE FLAGS SET. IF ANY BIT IS SET IN GROUP 1, 2, OR 3, GROUP 1 FORMAT (ERROR 4 OR 5) WILL BE USED; ANY SET IN GROUP 2 OR 3, GROUP 2 (ERROR 6) WILL BE USED; AND A FLAG SET IN GROUP 3 ONLY, GROUP 3 (ERROR 7) IS USED.

THE FORMAT TO BE USED IN THE CONTROLLING FACTOR IN HOW THE ERROR TRAP IS CHANGED IN THE MAIN CALL. IF GROUP 1 FORMAT IS USED THE ERROR TRAP WILL BE CHANGED TO ERROR 4 OR 5 (DEPENDING ON AVAILABILITY OF DRIVE STATUS), GROUP 2 FORMAT WILL BE ERROR 6, AND GROUP 3 WILL BE ERROR 7. ONLY THE LOW ORDER BYTE OF THE ERROR TRAP WILL BE ALTERED. THE SP WILL BE POINTING TO THE LOCATION THAT CONTAINS THE ERROR CALL TRAP.

IF THE STATUS IS READ FROM THE DRIVE WITH NO PROBLEM, ERROR 4 IS USED. IF ANY ERROR IS ENCOUNTERED READING STATUS, ERROR 5 IS USED. ERROR 5 INCLUDES A WARNING MESSAGE.

```

110$: CLR R4 ;CLEAR COUNTERS
      CLR R5
      MOV #STMP1,R0 ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
      MOV #STMP2,R1 ;WHERE ASCIZ ADDRESSES GO
      MOV #GPSUMF,R3 ;POINTERS TO GROUP SUMMARY FLAGS
      MOV #DF007A,(R0) ;PRESET FOR GRP3 ERR MESSAGE BUILD
      MOV #DH7N,(R1)
      MOV GRP3ER,-(SP) ;GET GROUP 3 ERRORS, PUT ON STACK
      JSR R4,BITCNT ;GO COUNT NUMBER AT ERRORS
      TST (SP) ;ANY ERRORS?
      BEQ 29$ ;NO - SKIP
      ADD (SP),R5 ;ADD IN ERROR TOTAL
      BIS #GRP3ST,(R3) ;SET BIT TO INDICATE GROUP 3 ERROR

29$: TST (SP)+ ;CLEAR OFF STACK
     TST STMP13 ;ERROR EXPECTED
     BEQ 31$ ;NO - SKIP
     MOV STMP12,-(SP) ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
     JSR R4,BITCNT ;COUNT NUMBER OF ERRORS.
     TST (SP) ;WERE THERE ANY
     BEQ 30$ ;NO - SKIP
     BIS #GP3NR,(R3) ;SET GROUP 3 NOT RECEIVED ERROR FLAG
     ADD (SP),R4 ;ADD COUNT TO TOTAL THESE

30$: TST (SP)+ ;CLEAR OFF STACK
31$: MOV GRP2ER,-(SP) ;GET GROUP 2 ERRORS FOR COUNTING
     JSR R4,BITCNT ;COUNT BITS
     TST (SP) ;ANY SET?
     BEQ 32$ ;NO - SKIP
     BIS #GRP2ST,(R3) ;SET FLAG FOR GROUP 2 ERRORS
  
```

8417	036500	061605		ADD	(SP),R5	;ADD INTO TOTAL
8418	036502	012710	055460	MOV	#DF006A,(R0)	;STORE ADDRESS FOR BUILDING REPORT
8419	036506	012711	001432	MOV	#DH6N,(R1)	
8420						
8421	036512	005726		32\$: TST	(SP)+	;CLEAR OFF STACK
8422	036514	005737	001250	TST	\$TMP13	;ANY EXPECTED ERRORS
8423	036520	001416		BEQ	34\$	;NO - SKIP
8424	036522	013746	001244	MOV	\$TMP11,-(SP)	;GET GROUP 2 NOT RECEIVED ERRORS
8425	036526	004437	033524	JSR	R4,BITCNT	;COUNT NUMBER OF BITS
8426	036532	005716		TST	(SP)	;ANY SET?
8427	036534	001407		BEQ	33\$	;NO - SKIP
8428	036536	052713	000020	BIS	#GP2NR,(R3)	;SET FLAG FOR GROUP 2 NOT RECEIVED
8429	036542	061604		ADD	(SP),R4	;ADD INTO TOTAL
8430	036544	012710	055460	MOV	#DF006A,(R0)	;STORE ADDRESS FOR BUILDING REPORT
8431	036550	012711	001432	MOV	#DH6N,(R1)	
8432						
8433	036554	005726		33\$: TST	(SP)+	;CLEAR OFF STACK
8434	036556	013746	044750	34\$: MOV	GRP1R,-(SP)	;GET GROUP 1 ERROR FLAGS
8435	036562	004437	033524	JSR	R4,BITCNT	;COUNT THE NUMBER OF BITS
8436	036566	005716		TST	(SP)	;ANY SET?
8437	036570	001407		BEQ	35\$	;NO - SKIP
8438	036572	052713	000001	BIS	#GRP1ST,(R3)	;SET FLAG FOR GROUP 1 ERRORS SET
8439	036576	061605		ADD	(SP),R5	;ADD INTO TOTAL
8440	036600	012710	055374	MOV	#DF004A,(R0)	;LOAD ADDRESS FOR BUILDING REPORT
8441	036604	012711	001412	MOV	#DH4N,(R1)	
8442	036610	005726		35\$: TST	(SP)+	;CLEAR OFF STACK
8443	036612	005737	001250	TST	\$TMP13	;ANY EXPECTED ERRORS?
8444	036616	001416		BEQ	60\$	;NO - SKIP
8445	036620	013746	001242	MOV	\$TMP10,-(SP)	;GET GROUP 1 NO RECEIVED ERROR
8446	036624	004437	033524	JSR	R4,BITCNT	;COUNT # OF BITS
8447	036630	005716		TST	(SP)	;ANY SET?
8448	036632	001407		BEQ	36\$	;NO - SKIP
8449	036634	052713	000010	BIS	#GP1NR,(R3)	;SET FLAG FOR GROUP 1 NOT RECEIVED
8450	036640	061604		ADD	(SP),R4	;ADD INTO TOTAL
8451	036642	012710	055374	MOV	#DF004A,(R0)	;LOAD ADDRESS FOR BUILDING REPORT
8452	036646	012711	001412	MOV	#DH4N,(R1)	
8453	036652	005726		36\$: TST	(SP)+	;CLEAR OFF STACK.
8454	036654	032713	000011	60\$: BIT	#GRP1ST!GP1NR,(R3)	;ANY GROUP 1 ERROR
8455	036660	001414		BEQ	52\$	;NO - SKIP
8456	036662	042713	040000	BIC	#DRSTER,(R3)	
8457	036666	004437	034236	JSR	R4,GETDRS	
8458	036672	000401		BR	51\$	;ERROR RETURN
8459	036674	000406		BR	52\$	;NO ERROR RETURN
8460	036676	012710	055424	51\$: MOV	#DF005A,(R0)	;CHANGE TO FORMAT 5 - STORE ADDRESS
8461	036702	012711	001422	MOV	#DH5N,(R1)	;FOR BUILDING REPORT
8462	036706	052713	040000	BIS	#DRSTER,(R3)	;SET DRIVE STATUS ERROR
8463	036712			52\$:		

8464				:		
8465				:		
8466				:		
8467				:		
8468				:		
8469				:		
8470				:		
8471				:		
8472				:		

THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED. THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER DF TABLE.

AT THIS TIME



```

8473      :      R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
8474      :      NOT EXPECTED OR
8475      :      THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
8476      :      R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
8477      :      DID NOT OCCUR.
8478      :      $TMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
8479      :      $TMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
8480      :      $TMP12 CONTAINS GROUP 3 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
8481      :      GRP1ER CONTAINS GROUP 1 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
8482      :      GRP2ER CONTAINS GROUP 2 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
8483      :      GRP3ER CONTAINS GROUP 3 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
8484      :      (R1)=$STMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
8485      :      DF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
8486      :      (R0)=$STMP1 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
8487      :      THE ERROR ITEM TABLE THAT MUST BE ALTERED TO PROVIDE A
8488      :      PROPER MESSAGE TO REPORT.
8489      :      (R3)=$GRSUMF CONTAIN FLAGS TO INDICATE WHICH OF THE GROUP
8490      :      ERROR FLAG FIELDS HAVE ERROR BITS STORED.
8491 036712 032777 020000 142220      BIT      #SW13,DSWR      ;IS REPORT INHIBITED?
8492 036720 001402      BEQ      37$      ;NO - SKIP
8493 036722 000137 037400      JMP      53$      ;ELSE EXIT
8494 036726 005737 001250 37$:      TST      $TMP13      ;WERE ERRORS EXPECTED?
8495 036732 001004      BNE      38$      ;YES - SKIP
8496      :
8497      :      IF NO ERRORS WERE EXPECTED, $TMP10,11, &12 ARE NOT MEANINGFUL
8498      :
8499 036734 012771 054377 000000      MOV      #DH007,D(R1) ;HEADER = ERROR IN OPERATION
8500 036742 000411      BR      39$
8501 036744 012771 054220 000000 38$:      MOV      #DH005,D(R1) ;PRESET HDRMSG = EXPECTED NOT SET
8502 036752 032713 000070      BIT      #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS?
8503 036756 001003      BNE      39$      ;YES - SKIP
8504 036760 012771 054315 000000      MOV      #DH006,D(R1) ;SET MESSAGE TO UNEXPECTED ERROR SET
8505 036766 013701 001540 39$:      MOV      T.CS1,R1 ;GET CS1
8506 036772 042701 177741      BIC      #177741,R1 ;CLEAR ALL BUT COMMAND
8507      :
8508 036776 016170 045074 000000      MOV      CMNDLB(R1),D(R0) ;MOVE ADDRESS OF COMMAND MESSAGE
8509      :      INTO REPORT
8510 037004 032713 000007      BIT      #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY GPR ERRORS?
8511 037010 001462      BEQ      46$      ;NO - SKIP GPR REPORT
8512      :
8513      :      PRINT ALL THE ERRORS CONTAINED IN THE GPR1,2,3ER(UNEXPECTED ERRORS)
8514      :
8515 037012 013701 044754      MOV      GRP3ER,R1 ;GET GROUP 3 ERROR FLAGS
8516 037016 012700 044774      MOV      #GRP3MS,R0 ;SET POINTER TO GRP3 ERROR MESSAGES
8517 037022 005037 001252 140$:      CLR      $TMP14 ;CLEAR GROUP PRINTING INDICATOR
8518 037026 012737 000021 001254 40$:      MOV      #17.,$TMP15 ;PRESET SHIFT COUNT
8519 037034 000241      CLC      ;CLEAR CARRY
8520 037036 006001 41$:      ROR      R1 ;ROTATE ERROR FLAGS
8521 037040 103406      BCS      42$      ;WAS BIT SHIFTED OUT SET?
8522 037042 062700 000002 141$:      ADD      #2,R0 ;NO - BUMP POINTER
8523 037046 005337 001254      DEC      $TMP15 ;DEC SHIFT COUNT
8524 037052 001371      BNE      41$      ;LOOP IF SHIFT NOT ZERO
8525 037054 000411      BR      44$
8526 037056 011037 037070 42$:      MOV      (R0),43$ ;GET ERROR MESSAGE ADDRESS FROM TABLE
8527 037062 104401 001273      TYPE      ,SCRLF ;TYPE CRLF
8528 037066 104401      TYPE      ;TYPE ERROR MESSAGE

```

```

8529 037070 000000      43$: .WORD      ;MESSAGE ADDRESS GOES HERE
8530 037072 005305      DEC      R5      ;DECEREMENT TOTAL ERROR COUNT.
8531 037074 001362      BNE     141$     ;LOOP IF ZERO
8532 037076 000427      BR      46$     ;ELSE EXIT GPR ERROR PRINT LOOP
8533
8534 037100 005713      44$: TST      (R3)   ;TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
8535 037102 100455      BMI     47$     ;YES - SKIP
8536 037104 005737 001252 TST     $TMP14  ;PRINTING GROUP 3?
8537 037110 001007      BNE     45$     ;NO -SKIP
8538 037112 013701 044752 MOV     GRP2ER,R1 ;ELSE SET TO GROUP 2, GET GRP2ER
8539 037116 012700 045014 MOV     #GRP2MS,RO ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
8540 037122 005237 001252 INC     $TMP14  ;BUMP TO INDICATE PRINTING GROUP 2
8541 037126 000737      BR      40$     ;GO RESTART PRINT LOOP
8542 037130 022737 000002 001252 45$: CMP     #2,$TMP14 ;PRINTING GROUP 1?
8543 037136 001407      BEQ     46$     ;YES - EXIT GPR ERROR PRINT LOOP.
8544 037140 013701 044750 MOV     GRP1ER,R1 ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
8545 037144 012700 045032 MOV     #GRP1MS,RO ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
8546 037150 005237 001252 INC     $TMP14  ;BUMP TO INDICATE PRINTING GROUP 1
8547 037154 000724      BR      40$     ;RESTART PRINT LOOP.
8548
8549 037156 005737 001250      46$: TST     $TMP13  ;EXPECTING ERRORS?
8550 037162 001452      BEQ     49$     ;NO - SKIP
8551
8552 ; PRINT ALL ERRORS CONTAINED IN $TMP10, 11, 12(NOT RECEIVED ERRORS)
8553
8554 037164 005713      TST     (R3)   ;TEST IF PRINTING NOT RECEIVED ERRORS
8555 037166 100423      BMI     47$     ;YES - SKIP
8556 037170 032713 000070 BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
8557 037174 001445      BEQ     49$     ;NO - SKIP
8558 037176 032713 000007 BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
8559 037202 001404      BEQ     146$    ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
8560 037204 104401 001273 TYPE    ,$CRLF  ;TYPE CRLF
8561 037210 104401 054315 TYPE    ,DH006  ;TYPE HEADER FOR PREVIOUS ERRORS
8562 037214 052737 100000 045072 146$: BIS     #REPNR,GPSUMF ;SET PRINTING NOT RECEIVED ERRORS SWITCH
8563 037222 010405      MOV     R4,R5  ;MOVE TOTAL ERRORS TO R5
8564 037224 013701 001246 MOV     $TMP12,R1 ;GET GRP3 NOT RECEIVED ERRORS
8565 037230 012700 044774 MOV     #GRP3MS,RO ;SET POINTER TO GROUP 3 MESSAGES
8566 037234 000672      BR      140$    ;GO START PRINT LOOP
8567 037236 005737 001252      47$: TST     $TMP14  ;PRINTING GROUP 3?
8568 037242 001007      BNE     48$     ;NO - SKIP
8569 037244 013701 001244 MOV     $TMP11,R1 ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
8570 037250 012700 045014 MOV     #GRP2MS,RO ;& SET POINTER TO GROUP 2 MESSAGE TABLE
8571 037254 005237 001252 INC     $TMP14  ;BUMP TO INDICATE GROUP 2 PRINTING
8572 037260 000662      BR      40$     ;GO START PRINT LOOP
8573 037262 022737 000002 001252 48$: CMP     #2,$TMP14 ;PRINTING GROUP 1?
8574 037270 001407      BEQ     49$     ;YES - EXIT LOOP
8575 037272 013701 001242 MOV     $TMP10,R1 ;SET POINTER TO GROUP 1 MESSAGE
8576 037276 012700 045032 MOV     #GRP1MS,RO ;TABLE AND GET GROUP 1 ERRORS.
8577 037302 005237 001252 INC     $TMP14  ;BUMP TO INDICATE GROUP 1 PRINTING
8578 037306 000647      BR      40$     ;START LOOP AGAIN.
8579
8580 037310 032713 000077      49$: BIT     #77,(R3) ;TEST IF ANY ERRORS TO BE REPORTED
8581 ; GRP1ST!GRP2ST!GRP3ST
8582 ; GP1NR!GP2NR!GP3NR
8583 037314 001004      BNE     61$     ;YES - SKIP
8584 037316 104414      RESREG 61$     ;ELSE EXIT

```

```

8585 037320 062716 000002          ADD    #2,(SP)          ;BUMP FOR GOOD RETURN
8586 037324 000002          RTI
8587
8588 037326 112776 000007 000000 61$:  MOVB  #7,(SP)          ;PRESET FOR GROUP 3 ERROR RETURN.
8589 037334 032713 000022          BIT   #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
8590 037340 001403          BEQ   50$              ;NO - SKIP
8591 037342 112776 000006 000000          MOVB  #6,(SP)          ;ELSE SET FOR GROUP 2 ERROR RETURN
8592
8593 037350 032713 000011          50$:  BIT   #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
8594 037354 001411          BEQ   53$              ;NO - SKIP
8595 037356 112776 000004 000000          MOVB  #4,(SP)          ;ELSE SET FOR GROUP 1 ERROR RETURN.
8596 037364 032713 040000          BIT   #DRSTER,(R3)    ;CHECK IF ERROR GETTING DRIVE STATUS
8597 037370 001403          BEQ   53$              ;NO - SKIP
8598 037372 112776 000005 000000          MOVB  #5,(SP)          ;ELSE CHANGE RETURN FORM GROUP 1
8599
8600 037400 013700 001302          53$:  MOV   $TESTN,R0        ;SET UP $ESCAPE TO FORCE
8601 037404 006300          ASL   R0               ;ABORT TO PRESENT TEST AFTER
8602 037406 016037 031040 001264          MOV   $$SWO8TB(R0),$ESCAPE ;ERROR IS REPORTED
8603 037414 162737 000002 001264          SUB   #2,$ESCAPE      ;BUT GO TO NEXT SCOPE STATEMENT
8604 037422 104414          RESREG
8605 037424 000002          RTI                    ;RETURN
8606

```

```

8607 ;*****
8608 ;SBTTL BAD SECTOR CHECK
8609 ;* THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
8610 ;* CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
8611 ;* THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
8612 ;* L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
8613 ;* THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
8614 ;* IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
8615 ;* IS PLACED ON THE STACK AFTER THE ENTIRE
8616 ;* FIELD IS SEARCHED.
8617 ;*
8618 ;* CALL: JSR R4,BDSRCK
8619 ;* <ADDRESS OF FIELD TO BE SEARCHED>
8620 ;*****
8621

```

```

8622 037426 012637 001236          BDSRCK: MOV   (SP)+,$TMP6      ;STORE OLD R4 CONTENTS
8623 037432 010437 001240          MOV   R4,$TMP7        ;GET RETURN ADDRESS
8624 037436 011404          MOV   (R4),R4         ;GET POINTER TO FIELD TO BE CHECKED
8625 037440 005037 001234          CLR   $TMP5           ;CLEAR A COUNTER
8626 037444 005714          1$:  TST   (R4)            ;TEST IF FIELD HAS NO (OR NO MORE) ENTRIES
8627 037446 100417          BMI   4$              ;YES - EXIT
8628 037450 023724 001614          CMP   L.DCYL,(R4)+    ;IS THIS ENTRY FOR THIS CYLINDER?
8629 037454 001012          BNE   3$              ;NO - SKIP
8630 037456 005204          INC   R4               ;BUMP TO TRACK
8631 037460 123714 001607          CMPB  L.DT,(R4)       ;IS ENTRY FOR THIS TRACK?
8632 037464 001005          BNE   2$              ;NO - SKIP
8633 037466 005046          CLR   -(SP)           ;CLEAR STACK LOCATION
8634 037470 114416          MOVB  -(R4),(SP)      ;PUT SECTOR NUMBER ON STACK
8635 037472 005237 001234          INC   $TMP5           ;BUMP COUNTER
8636 037476 000401          BR    3$              ;BRANCH
8637
8638 037500 005304          2$:  DEC   R4              ;DECREMENT POINTER TO WORD ALIGN
8639 037502 005724          3$:  TST   (R4)+          ;BUMP TO NEXT ENTRY
8640 037504 000757          BR    1$              ;TEST NEXT ENTRY

```

```

8641
8642 037506 013746 001234 4S: MOV STMP5,-(SP) ;PUT COUNT ON STACK
8643 037512 013746 001236 MOV STMP6,-(SP) ;PUT OLD R4 CONTENTS BACK ON STACK
8644 037516 013704 001240 MOV STMP7,R4 ;SET UP RETURN
8645 037522 005724 TST (R4)+ ;BUMP PAST PARAMETER
8646 037524 000204 RTS R4 ;RETURN
8647
    
```

8648 ;\*\*\*\*\*

8649 ;SBTTL DATA GENERATION AND COMPARE ROUTINE

```

8650 ;* CALLS: JSR R4,GENCOM
8651 ;* CONTROL WORD
    
```

```

8652 ;*
8653 ;* JSR R4,GENCOM
8654 ;* CONTROL WORD
8655 ;* LENGTH
    
```

```

8656 ;*
8657 ;* JSR R4,GENCOM
8658 ;* CONTROL WORD
8659 ;* RELOCATION CONSTANT
8660 ;* LENGTH
    
```

```

8661 ;* RETURN: RTS R4
8662 ;*
8663 ;* R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
8664 ;* THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
8665 ;* CONTROL WORD. THIS IS UNCONDITIONAL.
    
```

```

8666 ;*
8667 ;* THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
8668 ;* THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
8669 ;* IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
8670 ;* ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.
    
```

```

8671 ;* THE THIRD CALL IS IDENTICAL TO THE SECOND.
    
```

8672 ;\* DEFINITION OF CONTROL WORD:

- 8673 ;\* BIT 15 - DO COMPARE OPERATION OF Ibuff (SOURCE) TO Obuff
- 8674 ;\* (DESTINATION). EXPECTED VALUES ARE IN Obuff (DESTINATION).
- 8675 ;\* BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
- 8676 ;\* BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
- 8677 ;\* BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
- 8678 ;\* BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
- 8679 ;\* BIT 10 - CLEAR Ibuff TO PATTERN SELECTED.
- 8680 ;\* BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
- 8681 ;\* BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
- 8682 ;\* BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
- 8683 ;\* BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
- 8684 ;\* NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
- 8685 ;\* OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.

8686 ;\* EXPLANATION OF CALLS:

```

8687 ;* THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR
8688 ;* BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.
    
```

```

8689 ;*
8690 ;* THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED
8691 ;* FOR DATA GENERATION OR COMPARE AND FOR Ibuff INITIALIZATION.
    
```

```

8692 ;*
8693 ;* THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS
8694 ;*
8695 ;*
8696 ;*
    
```

8697  
8698  
8699  
8700  
8701  
8702  
8703  
8704  
8705  
8706  
8707  
8708  
8709  
8710  
8711  
8712  
8713  
8714  
8715  
8716  
8717  
8718  
8719  
8720  
8721  
8722  
8723  
8724  
8725  
8726  
8727  
8728  
8729  
8730  
8731  
8732  
8733

USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.

DESCRIPTION:

- THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:
- A. BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8) OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK ARE LISTED THEREIN (BIT 9).
  - B. COMPARE THE CONTENTS OF Ibuff AND Obuff (BIT 15). THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8) WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS MAY BE BUILT BEFORE THE COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
  - C. RESUME COMPARE OPERATION. IF A COMPARE OPERATION DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14 SET IN THE CONTROL WORD.
  - D. DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT. MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE DATA COMPARE WILL COMPARE THE CONTENTS OF Ibuff TO ANY AREA OF AVAILABLE MEMORY.

8734 037526  
8735 037526 010046  
8736 037530 010146  
8737 037532 010346  
8738 037534 010546  
8739 037536 012400  
8740 037540 012737 052744 001520  
8741 037546 032700 000200  
8742 037552 001005  
8743 037554 012737 052773 001520  
8744 037562 000137 040314  
8745 037566  
8746 037566 010446  
8747 037570 032700 001400  
8748 037574 001002  
8749 037576 000137 040076  
8750 037602 113701 001607  
8751 037606 013703 001614  
8752 037612 012705 000005

GENCOM:

```

MOV R0,-(SP)      ;;PUSH R0 ON STACK
MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV R3,-(SP)      ;;PUSH R3 ON STACK
MOV R5,-(SP)      ;;PUSH R5 ON STACK
MOV (R4)+,R0      ;;GET PARAMETER WORD
MOV #EM54,EM15N   ;;PRESET FOR HEADER COMPARE ERROR
BIT #BIT7,R0      ;;HEADER OPERATION SPECIFIED?
BNE 18$           ;;YES - SKIP
MOV #EM55,EM15N   ;;CHANGE FOR DATA COMPARE ERROR
JMP 17$           ;;ELSE JUMP TO DATA ROUTINE

18$:
MOV R4,-(SP)      ;;PUSH R4 ON STACK
BIT #BIT8!BIT9,R0 ;;MUST HEADERS BE BUILT?
BNE 19$           ;;YES - SKIP
JMP 11$           ;;ELSE JUMP TO HEADER COMPARE
19$:
MOVB L.DT,R1      ;;START HEADER BUILD ROUTINE
MOV L.DCYL,R3     ;;GET TRACK AND CYL
MOV #5,R5         ;;SET COUNT TO SHIFT TRACK FOR HDR WORD
    
```

8753											
8754	037616	006301		1\$:	ASL	R1					:SHIFT TRACK
8755	037620	005305			DEC	R5					:DECREMENT TRACK
8756	037622	001375			BNE	1\$					:LOOP UNTIL COUNT 0
8757											
8758	037624	012704	000026		MOV	#26,R4					:PRESET FOR 26 SECTOR MODE
8759	037630	032737	010000	001600	BIT	#CFMT,L.CS1					:IS IT 24 SECTOR MODE?
8760	037636	001404			BEQ	2\$					:NO - SKIP
8761	037640	012704	000024		MOV	#24,R4					:CHANGE COUNT FOR 24 SECTOR MODE
8762	037644	052701	001000		BIS	#BIT9,R1					:SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
8763											
8764	037650	052701	140000	2\$:	BIS	#BIT15:BIT14,R1					:SET BS BITS TO INDICATE GOOD SECTOR
8765	037654	012705	060446		MOV	#OBUFF,R5					:SET POINTER TO ADDRESS WHERE HEADERS GO
8766	037660	010325		3\$:	MOV	R3,(R5)+					:INSERT CYLINDER
8767	037662	010125			MOV	R1,(R5)+					:INSERT TRACK AND SECTOR
8768	037664	010337	001224		MOV	R3,\$TMP1					:CALCULATE HVRC WORD
8769	037670	010115			MOV	R1,(R5)					
8770	037672	040137	001224		BIC	R1,\$TMP1					
8771	037676	040315			BIC	R3,(R5)					
8772	037700	053725	001224		BIS	\$TMP1,(R5)+					:COMPLETE HVRC WORD INSERTION
8773											
8774	037704	005304			DEC	R4					:DECREMENT HEADER COUNT
8775	037706	001402			BEQ	4\$					:DONE? - YES, SKIP
8776	037710	005201			INC	R1					:BUMP SECTOR
8777	037712	000762			BR	3\$					:LOOP
8778											
8779	037714	032700	001000	4\$:	BIT	#BIT9,R0					:MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
8780	037720	001003			BNE	5\$					:YES - SKIP
8781	037722	005700		10\$:	TST	R0					:IS THIS A COMPARE OPERATION?
8782	037724	100464			BMI	11\$					:YES-GO DO HDR COMPARE
8783	037726	000534			BR	50\$					:ELSE GET OUT
8784											
8785	037730	013737	001640	037766	5\$:	MOV	BSF26P,6\$				:PRESET FOR BS FACTORY LIST
8786	037736	012737	100000	001224	MOV	#BIT15,\$TMP1					:SET BIT TO BE RESET IN BAD HEADER
8787	037744	032737	010000	001600	BIT	#CFMT,L.CS1					:IS THIS 26 SECTOR MODE?
8788	037752	001403			BEQ	8\$					:YES - SKIP
8789	037754	013737	001636	037766	MOV	BSF24P,6\$					:ELSE CHANGE FOR 24 SECTOR MODE
8790											
8791	037762	004437	037426		JSR	R4,BDSRCK					:GO CHECK FOR BAD SECTOR THIS ADDRESS
8792	037766	000000			.WORD	0					:POINTER TO FILE TO BE CHECKED GOES HERE
8793	037770	012605			MOV	(SP)+,R5					:GET # OF BAD SECTORS THIS PACK ADDRESS
8794	037772	001417			BEQ	9\$					:SKIP IF ZERO
8795											
8796	037774	011601		7\$:	MOV	(SP),R1					:GET 1ST BAD SECTOR NUMBER
8797	037776	006301			ASL	R1					:MULTIPLY SECTOR NUMBER BY 6 TO
8798	040000	006301			ASL	R1					:LOCATE SECTOR TO BE MARKED BAD
8799	040002	061601			ADD	(SP),R1					
8800	040004	062601			ADD	(SP)+,R1					
8801	040006	062701	000002		ADD	#2,R1					:ADD 2 FOR 2ND WORD THAT SECTOR
8802	040012	043761	001224	060446	BIC	\$TMP1,OBUFF(R1)					:CLEAR BIT FOR BAD SECTOR IN HDR
8803	040020	043761	001224	060450	BIC	\$TMP1,OBUFF+2(R1)					:CORRECT THE HVRC BIT
8804	040026	005305			DEC	R5					:DECREMENT BAD SECTOR COUNT
8805	040030	001361			BNE	7\$					:LOOP IF NOT ZERO
8806											
8807	040032	032737	100000	001224	9\$:	BIT	#BIT15,\$TMP1				:WERE WE DOING BS FACTORY LIST?
8808	040040	001730			BEQ	10\$					:NO - GO CHECK IF COMPARE MUST BE DONE

```

8809 040042 012737 040000 001224      MOV      #BIT14,$TMP1      ;ELSE SET BIT TO BE RESET IN BAD HDR
8810 040050 013737 001644 037766      MOV      B5526P,6$      ;PRESET POINTER FOR 26 SECTOR MODE
8811 040056 032737 010000 001600      BIT      #CFMT,L.CS1     ;TEST IF WE ARE DOING 26 SECTOR MODE
8812 040064 001736          BEQ      B$              ;YES - SKIP TO START CHECK
8813 040066 013737 001642 037766      MOV      B5524P,6$      ;CHANGE POINTER TOR 24 SECTOR MODE
8814 040074 000732          BR       B$              ;SKIP TO START CHECK
8815
8816
8817 040076 012701 000102          ;11$:   MOV      #102,R1        ;PRESET FOR 102 WORDS OF HEADER
8818 040102 032737 010000 001600      BIT      #CFMT,L.CS1     ;CHECK IF 26 SECTOR MODE
8819 040110 001402          BEQ      12$            ;YES - SKIP
8820 040112 012701 000074          MOV      #74,R1         ;CHANGE TO 74 WORDS OF HEADER
8821
8822 040116 012704 056446          ;12$:   MOV      #IBUFF,R4     ;SET START OF HEADERS TO BE COMPARED
8823 040122 012705 060446          MOV      #OBUFF,R5     ;SET START OF GOOD HEADERS
8824 040126 005003          CLR      R3             ;CLEAR COUNTER
8825 040130 032700 040000      BIT      #BIT14,R0      ;IS THIS A CONTINUATION OF EARLIER COMPARE
8826 040134 001412          BEQ      13$            ;NO - SKIP
8827 040136 013705 001664          ;28$:   MOV      DESHLD,R5   ;GET VALUES WHERE PREVIOUS CHECK STOPPED
8828 040142 013704 001666          MOV      SRCHLD,R4     ;DESTINATION AND SOURCE
8829 040146 013703 001670          MOV      WRDNUM,R3    ;WORD NUMBER IN ERROR
8830 040152 013701 001672          MOV      WRDCNT,R1    ;WORD COUNT LEFT IN COMPARE
8831 040156 005701          TST     R1              ;TEST IF WORD COUNT LEFT = 0
8832 040160 001417          BEQ      50$            ;YES - EXIT
8833
8834 040162 032700 030000          ;13$:   BIT      #BIT12!BIT13,R0 ;MEM MANAGE REQUIRED?
8835 040166 001402          BEQ      25$            ;NO - SKIP
8836 040170 005237 177572          INC      @#SRO          ;TURN IT ON
8837 040174 022425          ;25$:   CMP      (R4)+,(R5)+  ;COMPARE THE WORDS
8838 040176 001012          BNE      14$            ;SKIP IF NOT EQUAL
8839 040200 005203          INC      R3             ;BUMP WORD NUMBER IN ERROR
8840 040202 005301          DEC      R1             ;DEC WORD COUNT LEFT IN COMPARE
8841 040204 001373          BNE      25$            ;LOOP IF NOT ZERO
8842 040206 032700 030000      BIT      #BIT12!BIT13,R0 ;MEM MANAGE IN USE?
8843 040212 001402          BEQ      50$            ;NO - SKIP
8844 040214 005337 177572          DEC      @#SRO          ;TURN IT OFF
8845
8846 040220 012604          ;50$:   MOV      (SP)+,R4      ;;POP STACK INTO R4
8847 040222 000427          BR       16$
8848
8849 ; ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
8850
8851 040224 010537 001664          ;14$:   MOV      R5,DESHLD    ;STORE DESTINATION
8852 040230 010437 001666          MOV      R4,SRCHLD     ;SOURCE
8853 040234 014537 001202          MOV      -(R5),$REG10  ;LOAD GOOD WORD FOR REPORT
8854 040240 014437 001204          MOV      -(R4),$REG11  ;BAD WORD
8855 040244 010337 001206          MOV      R3,$REG12    ;WORD NUMBER
8856 040250 005301          DEC      R1             ;DEC COUNT LEFTFOR CONTINUATION
8857 040252 005203          INC      R3             ;BUMP BAD WORD NUMBER
8858 040254 010137 001672          MOV      R1,WRDCNT    ;STORE COUNT LEFT
8859 040260 010337 001670          MOV      R3,WRDNUM    ;WORD NUM IN ERROR
8860 040264 032700 030000      BIT      #BIT12!BIT13,R0 ;MEM MANAGE IS USE?
8861 040270 001402          BEQ      15$            ;NO - SKIP
8862 040272 005337 177572          DEC      @#SRO          ;TURN IT OFF
8863
8864 040276          ;15$:

```

```

8865 040276 012604          MOV    (SP)+,R4          ;;POP STACK INTO R4
8866 040300 005724          TST    (R4)+           ;ERROR RETURN
8867
8868 040302          16$:
8869 040302 012605          MOV    (SP)+,R5          ;;POP STACK INTO R5
8870 040304 012603          MOV    (SP)+,R3          ;;POP STACK INTO R3
8871 040306 012601          MOV    (SP)+,R1          ;;POP STACK INTO R1
8872 040310 012600          MOV    (SP)+,R0          ;;POP STACK INTO R0
8873 040312 000204          RTS    R4
8874
8875          ; DATA PATTERN PROCESSING ROUTINE
8876
8877 040314 032700 040000      17$:  BIT    #BIT14,R0      ;CONTINUE WITH COMPARE?
8878 040320 001402          BEQ    29$              ;NO - SKIP
8879 040322 010446          MOV    R4,-(SP)         ;STORE RETURN
8880 040324 000704          BR     28$              ;GO CONTINUE COMPARE
8881
8882 040326 012705 060446      29$:  MOV    #OBUFF,R5        ;GET DESTINATION
8883 040332 012703 056446          MOV    #IBUFF,R3        ;GET SOURCE
8884 040336 032700 030000          BIT    #BIT12!BIT13,R0 ;USE MEM MANAGE?
8885 040342 001412          BEQ    21$              ;NO - SKIP
8886
8887 040344 012437 172354          MOV    (R4)+,#KIPAR6    ;LOAD PAR FOR RELOCATION
8888 040350 032700 010000          BIT    #BIT12,R0        ;RELOCATE SOURCE?
8889 040354 001403          BEQ    20$              ;NO - SKIP
8890 040356 012705 140070          MOV    #140070,R5       ;SET DESTINATION TO USE PAR6 + OFFSET
8891 040362 000402          BR     20$              ;SKIP
8892 040364 012703 140070      20$:  MOV    #140070,R3       ;SET SOURCE TO USE PAR6 + OFFSET
8893
8894 040370 012401          21$:  MOV    (R4)+,R1         ;STORE COUNT
8895 040372 010446          MOV    R4,-(SP)         ;STORE RETURN
8896 040374 010304          MOV    R3,R4            ;PUT IN IBUFF POINTER
8897 040376 005003          CLR    R3               ;CLEAR R3 FOR WORD NUMBER COUNTER
8898 040400 032700 000077          BIT    #77,R0           ;ANY DATA PATTERN SPECIFIED?
8899 040404 001666          BEQ    13$              ;NO - GO DO COMPARE
8900
8901          ; START OF GENERATION ROUTINE
8902
8903 040406 010537 001664          MOV    R5,DESHLD        ;STORE PARAMETERS FOR COMPARE
8904 040412 010437 001666          MOV    R4,SRCHLD
8905 040416 010337 001670          MOV    R3,WRDNUM
8906 040422 010137 001672          MOV    R1,WRDCNT
8907
8908          ;
8909          ; CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
8910          ; MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
8911          ; WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
8912 040426 032700 030000          BIT    #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
8913 040432 001402          BEQ    33$              ;NO - SKIP
8914 040434 005237 177572          INC    #SRO             ;TURN IT ON
8915 040440 032700 002000      33$:  BIT    #BIT10,R0        ;GENERATE PATTERN IN IBUFF?
8916 040444 001403          BEQ    32$              ;NO - SKIP
8917 040446 010446          MOV    R4,-(SP)         ;ELSE SWAP R4 AND R5
8918 040450 010504          MOV    R5,R4
8919 040452 012605          MOV    (SP)+,R5
8920
    
```



8921	040454	122700	000001	32\$:	CMPB	#1,RO	;PATTERN 1 (ALL ZEROS)?
8922	040460	001004			BNE	55\$	;NO - SKIP
8923	040462	005025		30\$:	CLR	(R5)+	;CLEAR WORD IN BUFF
8924	040464	005301			DEC	R1	;DEC WORD COUNT
8925	040466	001375			BNE	30\$	;LOOP UNTIL WORD COUNT ZERO
8926	040470	000550			BR	22\$	;EXIT BUILD
8927							
8928	040472	122700	000007	55\$:	CMPB	#7,RO	;PATTERN 7 (ALL ONES)?
8929	040476	001005			BNE	56\$	;NO - SKIP
8930	040500	012725	177777	31\$:	MOV	#-1,(R5)+	;LOAD WORD IN BUFF
8931	040504	005301			DEC	R1	;DEC WORD COUNT
8932	040506	001374			BNE	31\$	;LOOP UNTIL WORD COUNT ZERO
8933	040510	000540			BR	22\$	;EXIT BUILD
8934							
8935	040512	122700	000002	56\$:	CMPB	#2,RO	;PATTERN 2 SET UP
8936	040516	001003			BNE	57\$	
8937	040520	012703	044120		MOV	#PAT02,R3	
8938	040524	000504			BR	70\$	
8939							
8940	040526	122700	000003	57\$:	CMPB	#3,RO	;PATTERN 3 SET UP
8941	040532	001003			BNE	58\$	
8942	040534	012703	044160		MOV	#PAT03,R3	
8943	040540	000476			BR	70\$	
8944							
8945	040542	122700	000004	58\$:	CMPB	#4,RO	;PATTERN 4 SET UP
8946	040546	001003			BNE	59\$	
8947	040550	012703	044220		MOV	#PAT04,R3	
8948	040554	000470			BR	70\$	
8949							
8950	040556	122700	000005	59\$:	CMPB	#5,RO	;PATTERN 5 SET UP
8951	040562	001003			BNE	60\$	
8952	040564	012703	044260		MOV	#PAT05,R3	
8953	040570	000462			BR	70\$	
8954							
8955	040572	122700	000006	60\$:	CMPB	#6,RO	;PATTERN 6 SET UP
8956	040576	001003			BNE	61\$	
8957	040600	012703	044320		MOV	#PAT06,R3	
8958	040604	000454			BR	70\$	
8959							
8960	040606	122700	000010	61\$:	CMPB	#10,RO	;PATTERN 10 SET UP
8961	040612	001003			BNE	62\$	
8962	040614	012703	044360		MOV	#PAT10,R3	
8963	040620	000446			BR	70\$	
8964							
8965	040622	122700	000011	62\$:	CMPB	#11,RO	;PATTERN 11 SET UP
8966	040626	001003			BNE	63\$	
8967	040630	012703	044420		MOV	#PAT11,R3	
8968	040634	000440			BR	70\$	
8969							
8970	040636	122700	000012	63\$:	CMPB	#12,RO	;PATTERN 12 SET UP
8971	040642	001003			BNE	64\$	
8972	040644	012703	044460		MOV	#PAT12,R3	
8973	040650	000432			BR	70\$	
8974							
8975	040652	122700	000013	64\$:	CMPB	#13,RO	;PATTERN 13 SET UP
8976	040656	001003			BNE	65\$	

8977	040660	012703	044520		MOV	#PAT13,R3	
8978	040664	000424			BR	70\$	
8979							
8980	040666	122700	000014	65\$:	CMPB	#14,R0	;PATTERN 14 SET UP
8981	040672	001003			BNE	66\$	
8982	040674	012703	044560		MOV	#PAT14,R3	
8983	040700	000416			BR	70\$	
8984							
8985	040702	122700	000015	66\$:	CMPB	#15,R0	;PATTERN 15 SET UP
8986	040706	001003			BNE	67\$	
8987	040710	012703	044620		MOV	#PAT15,R3	
8988	040714	000410			BR	70\$	
8989							
8990	040716	122700	000016	67\$:	CMPB	#16,R0	;PATTERN 16 SET UP
8991	040722	001003			BNE	68\$	
8992	040724	012703	044660		MOV	#PAT16,R3	
8993	040730	000402			BR	70\$	
8994							
8995	040732	012703	044660	68\$:	MOV	#PAT16,R3	;SET UP FOR 16
8996							
8997	040736	032700	004000	70\$:	BIT	#BIT11,R0	;FIRST WORD REPEAT?
8998	040742	001020			BNE	73\$	;YES - SKIP
8999	040744	010446			MOV	R4,-(SP)	;STORE R4
9000	040746	010046			MOV	R0,-(SP)	;STORE R0
9001	040750	012700	000020		MOV	#16,R0	;PRESET COUNT FOR PATTERN LENGTH
9002	040754	010504			MOV	R5,R4	;STORE START OF BUFF
9003							
9004	040756	012325		71\$:	MOV	(R3)+,(R5)+	;MOV WORD TO BUFF
9005	040760	005301			DEC	R1	;DEC WORD COUNT
9006	040762	001405			BEQ	74\$	;EXIT IF ZERO
9007	040764	005300			DEC	R0	;DEC PAT LENGTH COUNT
9008	040766	001373			BNE	71\$	;LOOP IF NOT ZERO
9009							
9010	040770	012425		72\$:	MOV	(R4)+,(R5)+	;REPEAT PATTERN IN BUFFER
9011	040772	005301			DEC	R1	;DEC WORD COUNT
9012	040774	001375			BNE	72\$	;LOOP UNTIL WORD COUNT ZERO
9013							
9014	040776	012600		74\$:	MOV	(SP)+,R0	;RESTORE R0
9015	041000	012604			MOV	(SP)+,R4	;RESTORE R4
9016	041002	000403			BR	22\$	;EXIT BUILD
9017							
9018	041004	011325		73\$:	MOV	(R3),(R5)+	;MOV THE SAME WORD INTO BUFFER
9019	041006	005301			DEC	R1	;DEC WORD COUNT
9020	041010	001375			BNE	73\$	;LOOP UNTIL ZERO
9021							
9022	041012	032700	030000	22\$:	BIT	#BIT12:BIT13,R0	;MEMORY MANAGEMENT REQUIRED?
9023	041016	001402			BEQ	34\$	;NO - SKIP
9024	041020	005337	177572		DEC	#SRO	;TURN OFF MEM MANAGEMENT
9025	041024	005700		34\$:	TST	R0	;IS COMPARE REQUIRED?
9026	041026	100012			BPL	23\$	;NO - SKIP
9027	041030	013705	001664		MOV	DESHLD,R5	;RESTORE COMPARE PARAMETERS
9028	041034	013704	001666		MOV	SRCHLD,R4	
9029	041040	013703	001670		MOV	WRDNUM,R3	
9030	041044	013701	001672		MOV	WRDCNT,R1	
9031	041050	000137	040162		JMP	13\$	
9032	041054			23\$:			;GO START COMPARE

```

9033 041054 012604          MOV      (SP)+,R4      ;:POP STACK INTO R4
9034 041056 000137 040302  JMP      16$          ;:GO TO EXIT
9035
9036 ;:*****
9037 ;:SBTTL PHASE LOCK LOOP CLOCK ADJUSTMENT ROUTINE
9038 ;:* THIS ROUTINE IS ENTERED VIA A START AT LOCATION 22C(8). THE
9039 ;:* PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
9040 ;:* PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
9041 ;:* THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
9042 ;:* RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
9043 ;:* SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
9044 ;:*
9045 ;:* THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
9046 ;:*
9047 041062 104401 047626  ADJCLK: TYPE      ,OPR019      ;:TYPE ADJUSTMENT INSTRUCTIONS
9048
9049 041066 104416          TSSINIT          ;:CLEAR SUBSYSTEM
9050 041070 104003  ERROR      3          ;:BAD INIT ERROR
9051
9052 041072 004437 033324  JSR      R4,LRLOAD      ;:LOAD "L" REGS
9053 041076 000101  SELDRV          ;:SELDV
9054 041100 000000  0          ;:0 WORDS
9055 041102 000000  0          ;:0 IS BUFF ADDRESS
9056 041104 000      .BYTE      0          ;:SECTOR 0
9057 041105 000      .BYTE      0          ;:TRACK 0
9058 041106 000000  0          ;:CYLINDER 0
9059
9060 041110 104423  TWAT16          ;:WAIT FOR INTERRUPT
9061 041112 104002  ERROR      2          ;:TO SLOW/NOT RECEIVED ERROR
9062
9063 041114 104421  TCHKOP          ;:CHECK OPERATION FOR ANY ERRORS
9064 041116 104004  ERROR      4 ;OR 5, 6, 7 ;:REPORT ALL ERRORS
9065
9066 041120 012762 000040 000026 1$: MOV      #DMD,RKMR1(R2) ;:SET DIAG MODE
9067 041126 012701 000014          MOV      #12.,R1      ;:SET A COUNT
9068 041132 005301          DEC      R1          ;:DEC COUNT
9069 041134 001376          BNE     2$          ;:LOOP UNTIL ZERO
9070 041136 012762 000000 000026  MOV      #0,RKMR1(R2) ;:CLEAR MR1
9071 041144 012701 000014          MOV      #12.,R1      ;:SET COUNT
9072 041150 005301          DEC      R1          ;:DEC COUNT
9073 041152 001376          BNE     3$          ;:LOOP UNTIL ZERO
9074 041154 000761          BR      1$          ;:RESTART LOOP
9075
9076 ;:SBTTL TYPE ROUTINE
9077
9078 ;:*****
9079 ;:*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
9080 ;:*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
9081 ;:*NOTE1: SNUL contains the character to be used as the filler character.
9082 ;:*NOTE2: SFILLS contains the number of filler characters required.
9083 ;:*NOTE3: SFILLC contains the character to fill after.
9084 ;:*
9085 ;:*CALL:
9086 ;:*1) USING A TRAP INSTRUCTION
9087 ;:* TYPE      ,MESADR      ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
9088 ;:*OR

```

```

9089          : *      TYPE
9090          : *      MESADR
9091          : *
9092
9093 041156 105737 001157 $TYPE: TSTB  STPFLG  ;; IS THERE A TERMINAL?
9094 041162 100002          BPL      1$      ;; BR IF YES
9095 041164 000000          HALT          ;; HALT HERE IF NO TERMINAL
9096 041166 000430          BR      3$      ;; LEAVE
9097 041170 010046          1$:  MOV    RO,-(SP)  ;; SAVE RO
9098 041172 017600 000002  MOV    22(SP),RO  ;; GET ADDRESS OF ASCIZ STRING
9099 041176 122737 000001 001316  CMPB  #APTENV,$ENV  ;; RUNNING IN APT MODE
9100 041204 001011          BNE    62$     ;; NO, GO CHECK FOR APT CONSOLE
9101 041206 132737 000100 001317  BITB  #APTPOOL,$ENVM  ;; SPOOL MESSAGE TO APT
9102 041214 001405          BEQ    62$     ;; NO, GO CHECK FOR CONSOLE
9103 041216 010037 041226  MOV    RO,61$    ;; SETUP MESSAGE ADDRESS FOR APT
9104 041222 004737 031270  JSR   PC,$ATY3  ;; SPOOL MESSAGE TO APT
9105 041226 000000          .WORD  0      ;; MESSAGE ADDRESS
9106 041230 132737 000040 001317  62$: BITB  #APTCSUP,$ENVM  ;; APT CONSOLE SUPPRESSED
9107 041236 001003          BNE    60$     ;; YES, SKIP TYPE OUT
9108 041240 112046          2$:  MOVB  (RO)+,-(SP)  ;; PUSH CHARACTER TO BE TYPED ONTO STACK
9109 041242 001005          BNE    4$      ;; BR IF IT ISN'T THE TERMINATOR
9110 041244 005726          TST   (SP)+    ;; IF TERMINATOR POP IT OFF THE STACK
9111 041246 012600          60$: MOV    (SP)+,RO  ;; RESTORE RO
9112 041250 062716 000002  3$:  ADD   #2,(SP)  ;; ADJUST RETURN PC
9113 041254 000002          RTI          ;; RETURN
9114 041256 122716 000011  4$:  CMPB  #HT,(SP)  ;; BRANCH IF <HT>
9115 041262 001430          BEQ    8$      ;; BRANCH IF NOT <CRLF>
9116 041264 122716 000200  CMPB  #CRLF,(SP)  ;; BRANCH IF NOT <CRLF>
9117 041270 001006          BNE    5$      ;; POP <CR><LF> EQUIV
9118 041272 005726          TST   (SP)+    ;; TYPE A CR AND LF
9119 041274 104401          TYPE          ;; TYPE A CR AND LF
9120 041276 001273          $CRLF
9121 041300 105037 041434  CLRB  $CHARCNT  ;; CLEAR CHARACTER COUNT
9122 041304 000755          BR    2$      ;; GET NEXT CHARACTER
9123 041306 004737 041370  5$:  JSR   PC,$TYPEC  ;; GO TYPE THIS CHARACTER
9124 041312 123726 001156  6$:  CMPB  $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
9125 041316 001350          BNE    2$      ;; IF NO GO GET NEXT CHAR.
9126 041320 013746 001154  MOV    $NULL,-(SP)  ;; GET # OF FILLER CHARS. NEEDED
9127          ;; AND THE NULL CHAR.
9128 041324 105366 000001  7$:  DECB  1(SP)    ;; DOES A NULL NEED TO BE TYPED?
9129 041330 002770          BLT   6$      ;; BR IF NO--GO POP THE NULL OFF OF STACK
9130 041332 004737 041370  JSR   PC,$TYPEC  ;; GO TYPE A NULL
9131 041336 105337 041434  DECB  $CHARCNT  ;; DO NOT COUNT AS A COUNT
9132 041342 000770          BR    7$      ;; LOOP
9133
9134          ;HORIZONTAL TAB PROCESSOR
9135
9136 041344 112716 000040  8$:  MOVB  #' ,(SP)  ;; REPLACE TAB WITH SPACE
9137 041350 004737 041370  9$:  JSR   PC,$TYPEC  ;; TYPE A SPACE
9138 041354 132737 000007 041434  BITB  #7,$CHARCNT  ;; BRANCH IF NOT AT
9139 041362 001372          BNE    9$      ;; TAB STOP
9140 041364 005726          TST   (SP)+    ;; POP SPACE OFF STACK
9141 041366 000724          BR    2$      ;; GET NEXT CHARACTER
9142 041370 105777 137554  $TYPEC: TSTB  2$TPS  ;; WAIT UNTIL PRINTER IS READY
9143 041374 100375          BPL   $TYPEC
9144 041376 116677 000002 137546  MOVB  2(SP),2$TPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.

```

```

9145 041404 122766 000015 000002          CMPB   #CR,2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
9146 041412 001003          BNE    1$            ;; BRANCH IF NO
9147 041414 105037 041434          CLRB   $CHARCNT     ;; YES--CLEAR CHARACTER COUNT
9148 041420 000406          BR     $TYPEX       ;; EXIT
9149 041422 122766 000012 000002 1$:  CMPB   #LF,2(SP)     ;; IS CHARACTER A LINE FEED?
9150 041430 001402          BEQ    $TYPEX       ;; BRANCH IF YES
9151 041432 105227          INCB   (PC)+        ;; COUNT THE CHARACTER
9152 041434 000000          $CHARCNT: .WORD    0 ;; CHARACTER COUNT STORAGE
9153 041436 000207          $TYPEX: RTS        PC
9154
9155          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9156
9157          ;; *****
9158          ;; *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
9159          ;; *SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
9160          ;; *NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
9161          ;; *BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
9162          ;; *REPLACED WITH SPACES.
9163          ;; *CALL:
9164          ;; *      MOV     NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
9165          ;; *      TYPDS          ;; GO TO THE ROUTINE
9166
9167          $TYPDS:
9168          041440 010046          MOV    R0,-(SP)     ;; PUSH R0 ON STACK
9169          041442 010146          MOV    R1,-(SP)     ;; PUSH R1 ON STACK
9170          041444 010246          MOV    R2,-(SP)     ;; PUSH R2 ON STACK
9171          041446 010346          MOV    R3,-(SP)     ;; PUSH R3 ON STACK
9172          041450 010546          MOV    R5,-(SP)     ;; PUSH R5 ON STACK
9173          041452 012746 020200          MOV    #20200,-(SP) ;; SET BLANK SWITCH AND SIGN
9174          041456 016605 000020          MOV    20(SP),R5    ;; GET THE INPUT NUMBER
9175          041462 100004          BPL    1$            ;; BR IF INPUT IS POS.
9176          041464 005405          NEG    R5            ;; MAKE THE BINARY NUMBER POS.
9177          041466 112766 000055 000001 1$:  MOV    #'-,1(SP)    ;; MAKE THE ASCII NUMBER NEG.
9178          041474 005000          CLR    R0            ;; ZERO THE CONSTANTS INDEX
9179          041476 012703 041654          MOV    #$DBLK,R3    ;; SETUP THE OUTPUT POINTER
9180          041502 112723 000040          MOV    #' ,(R3)+    ;; SET THE FIRST CHARACTER TO A BLANK
9181          041506 005002          2$:  CLR    R2            ;; CLEAR THE BCD NUMBER
9182          041510 016001 041644          MOV    $DTBL(R0),R1 ;; GET THE CONSTANT
9183          041514 160105          3$:  SUB    R1,R5        ;; FORM THIS BCD DIGIT
9184          041516 002402          BLT    4$            ;; BR IF DONE
9185          041520 005202          INC    R2            ;; INCREASE THE BCD DIGIT BY 1
9186          041522 000774          BR     3$
9187          041524 060105          4$:  ADD    R1,R5        ;; ADD BACK THE CONSTANT
9188          041526 005702          TST    R2            ;; CHECK IF BCD DIGIT=0
9189          041530 001002          BNE    5$            ;; FALL THROUGH IF 0
9190          041532 105716          TSTB   (SP)         ;; STILL DOING LEADING 0'S?
9191          041534 100407          BMI    7$            ;; BR IF YES
9192          041536 106316          5$:  ASLB   (SP)         ;; MSD?
9193          041540 103003          BCC    6$            ;; BR IF NO
9194          041542 116663 000001 177777          MOV    1(SP),-1(R3) ;; YES--SET THE SIGN
9195          041550 052702 000060          6$:  BIS    #'0,R2      ;; MAKE THE BCD DIGIT ASCII
9196          041554 052702 000040          7$:  BIS    #' ,R2      ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
9197          041560 110223          MOV    R2,(R3)+    ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
9198          041562 005720          TST    (R0)+        ;; JUST INCREMENTING
9199          041564 020027 000010          CMP    R0,#10      ;; CHECK THE TABLE INDEX
9200          041570 002746          BLT    2$          ;; GO DO THE NEXT DIGIT

```

```

9201 041572 003002          BGT      8$          ;;GO TO EXIT
9202 041574 010502          MOV      R5,R2       ;;GET THE LSD
9203 041576 000764          BR       6$          ;;GO CHANGE TO ASCII
9204 041600 105726          8$:      TSTB      (SP)+  ;;WAS THE LSD THE FIRST NON-ZERO?
9205 041602 100003          BPL      9$          ;;BR IF NO
9206 041604 116663 177777 177776 9$:      MOV      -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYFING
9207 041612 105013          CLRB     (R3)        ;;SET THE TERMINATOR
9208 041614 012605          MOV      (SP)+,R5    ;;POP STACK INTO R5
9209 041616 012603          MOV      (SP)+,R3    ;;POP STACK INTO R3
9210 041620 012602          MOV      (SP)+,R2    ;;POP STACK INTO R2
9211 041622 012601          MOV      (SP)+,R1    ;;POP STACK INTO R1
9212 041624 012600          MOV      (SP)+,R0    ;;POP STACK INTO R0
9213 041626 104401 041654 000002 000004 TYPE    $DBLK      ;;NOW TYPE THE NUMBER
9214 041632 016666          MOV      2(SP),4(SP) ;;ADJUST THE STACK
9215 041640 012616          MOV      (SP)+,(SP)
9216 041642 000002          RTI
9217 041644 023420          $DTBL: 1000.        ;;RETURN TO USER
9218 041646 001750          1000.
9219 041650 000144          100.
9220 041652 000012          10.
9221 041654 000004          $DBLK: .BLKW 4
          .SBTTL BINARY TO OCTAL (ASCII) AND TYPE

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
9247 041664 017646 000000 042107 $TYPOS: MOV      2(SP),-(SP)  ;;PICKUP THE MODE
9248 041670 116637 000001 042107 MOV      1(SP),$OFILL ;;LOAD ZERO FILL SWITCH
9249 041676 112637 042111          MOV      (SP)+,$OMODE+1 ;;NUMBER OF DIGITS TO TYPE
9250 041702 062716 000002          ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
9251 041706 000406          BR       $TYPON
9252 041710 112737 000001 042107 $TYPOC: MOV      #1,$OFILL ;;SET THE ZERO FILL SWITCH
9253 041716 112737 000006 042111 MOV      #6,$OMODE+1 ;;SET FOR SIX(6) DIGITS
9254 041724 112737 000005 042106 $TYPON: MOV      #5,$OCNT  ;;SET THE ITERATION COUNT
9255 041732 010346          MOV      R3,-(SP)    ;;SAVE R3
9256 041734 010446          MOV      R4,-(SP)    ;;SAVE R4

```

```

9257 041736 010546          MOV      R5, -(SP)          ;; SAVE R5
9258 041740 113704 042111  MOVVB   $OMODE+1,R4      ;; GET THE NUMBER OF DIGITS TO TYPE
9259 041744 005404          NEG      R4
9260 041746 062704 000006  ADD      #6,R4            ;; SUBTRACT IT FOR MAX. ALLOWED
9261 041752 110437 042110  MOVVB   R4,$OMODE        ;; SAVE IT FOR USE
9262 041756 113704 042107  MOVVB   $OFILL,R4        ;; GET THE ZERO FILL SWITCH
9263 041762 016605 000012  MOV      12(SP),R5        ;; PICKUP THE INPUT NUMBER
9264 041766 005003          CLR      R3            ;; CLEAR THE OUTPUT WORD
9265 041770 006105          1$:    ROL      R5            ;; ROTATE MSB INTO "C"
9266 041772 000404          BR      3$
9267 041774 006105          2$:    ROL      R5            ;; GO DO MSB
9268 041776 006105          ROL      R5            ;; FORM THIS DIGIT
9269 042000 006105          ROL      R5
9270 042002 010503          MOV      R5,R3
9271 042004 006103          3$:    ROL      R3            ;; GET LSB OF THIS DIGIT
9272 042006 105337 042110  DECB   $OMODE            ;; TYPE THIS DIGIT?
9273 042012 100016          BPL     7$              ;; BR IF NO
9274 042014 042703 177770  BIC     #177770,R3        ;; GET RID OF JUNK
9275 042020 001002          BNE     4$              ;; TEST FOR 0
9276 042022 005704          TST     R4              ;; SUPPRESS THIS 0?
9277 042024 001403          BEQ     5$              ;; BR IF YES
9278 042026 005204          4$:    INC      R4            ;; DON'T SUPPRESS ANYMORE 0'S
9279 042030 052703 000060  BIS     #'0,R3            ;; MAKE THIS DIGIT ASCII
9280 042034 052703 000040  5$:    BIS     #' ,R3        ;; MAKE ASCII IF NOT ALREADY
9281 042040 110337 042104  MOVVB   R3,$$            ;; SAVE FOR TYPING
9282 042044 104401 042104  TYPE    .8$              ;; GO TYPE THIS DIGIT
9283 042050 105337 042106  7$:    DECB   $OCNT        ;; COUNT BY 1
9284 042054 003347          BGT     2$              ;; BR IF MORE TO DO
9285 042056 002402          BLT     6$              ;; BR IF DONE
9286 042060 005204          INC     R4              ;; INSURE LAST DIGIT ISN'T A BLANK
9287 042062 000744          BR      2$              ;; GO DO THE LAST DIGIT
9288 042064 012605          6$:    MOV     (SP)+,R5        ;; RESTORE R5
9289 042066 012604          MOV     (SP)+,R4        ;; RESTORE R4
9290 042070 012603          MOV     (SP)+,R3        ;; RESTORE R3
9291 042072 016666 000002 000004  MOV     2(SP),4(SP)      ;; SET THE STACK FOR RETURNING
9292 042100 012616          MOV     (SP)+,(SP)
9293 042102 000002          RTI
9294 042104 000          8$:    .BYTE 0            ;; RETURN
9295 042105 000          .BYTE 0            ;; STORAGE FOR ASCII DIGIT
9296 042106 000          $OCNT: .BYTE 0        ;; TERMINATOR FOR TYPE ROUTINE
9297 042107 000          $OFILL: .BYTE 0       ;; OCTAL DIGIT COUNTER
9298 042110 000000          $OMODE: .WORD 0        ;; ZERO FILL SWITCH
9299          .SBTTL TTY INPUT ROUTINE ;; NUMBER OF DIGITS TO TYPE
9300
9301          ;; *****
9302          .ENABL LSB
9303 042112 000000          $TKCNT: .WORD 0        ;; NUMBER OF ITEMS IN QUEUE
9304 042114 000000          $TKQIN: .WORD 0        ;; INPUT POINTER
9305 042116 000000          $TKQOUT: .WORD 0       ;; OUTPUT POINTER
9306 042120 000001          $TKQSRT: .BLKB 1       ;; TTY KEYBOARD QUEUE
9307          $TKQEND=.
9308          .EVEN
9309
9310          ;*TK INITIALIZE ROUTINE
9311          ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
9312          ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT

```

```

9313
9314
9315
9316
9317
9318 042122 005037 042112
9319 042126 012737 042120 042114
9320 042134 013737 042114 042116
9321 042142 012737 042172 000060
9322 042150 012737 000200 000062
9323 042156 005777 136764
9324 042162 012777 000100 136754
9325 042170 000207
9326
9327
9328
9329
9330
9331
9332
9333
9334 042172 117746 136750
9335 042176 042716 177600
9336 042202 021627 000003
9337 042206 001007
9338 042210 104401 043306
9339 042214 004737 042122
9340 042220 005726
9341 042222 000137 001736
9342 042226 021627 000007
9343 042232 001004
9344 042234 022737 000176 001140
9345 042242 001500
9346
9347 042244
9348 042244 022737 000001 042112
9349 042252 001004
9350 042254 104401 001266
9351 042260 005726
9352 042262 000451
9353 042264 021627 000023
9354 042270 001021
9355 042272 005077 136646
9356 042276 005726
9357 042300 105777 136640
9358 042304 100375
9359 042306 117746 136634
9360 042312 042716 177600
9361 042316 022627 000021
9362 042322 001366
9363 042324 012777 000100 136612
9364 042332 000002
9365 042334 005237 042112
9366 042340 021627 000140
9367 042344 002405
9368 042346 021627 000175

; *CALL:
; * JSR PC,STKINT
; * RETURN

$TKINT: CLR $TKCNT ;: CLEAR COUNT OF ITEMS IN QUEUE
MOV $TKQSRT,$TKQIN ;: MOVE THE STARTING ADDRESS OF THE
MOV $TKQIN,$TKQOUT ;: QUEUE INTO THE INPUT & OUTPUT POINTERS.
MOV $TKSRV,$TKVEC ;: INITIALIZE THE KEYBOARD VECTOR
MOV #200,$TKVEC+2 ;: "BR" LEVEL 4
TST $TKB ;: CLEAR DONE FLAG
MOV #100,$TKS ;: ENABLE TTY KEYBOARD INTERRUPT
RTS PC ;: RETURN TO CALLER

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

$TKSRV: MOV $TKB,-(SP) ;: PICKUP THE CHARACTER
BIC #↑C177,(SP) ;: STRIP THE JUNK
CMP (SP),#3 ;: IS IT A CONTROL C?
BNE 1$ ;: BRANCH IF NO
TYPE $CNTLC ;: TYPE A CONTROL-C (↑C)
JSR PC,$TKINT ;: INIT THE KEYBOARD
TST (SP)+ ;: CLEAN UP STACK
JMP START ;: CONTROL C RESTART
1$: CMP (SP),#7 ;: IS IT A CONTROL G?
BNE 2$ ;: BRANCH IF NO
CMP #SWREG,SWR ;: IS SOFT-SWR SELECTED?
BEQ 6$ ;: GO TO SWR CHANGE

2$:
CMP #1,$TKCNT ;: IS THE QUEUE FULL?
BNE 3$ ;: BRANCH IF NO
TYPE $BELL ;: RING THE TTY BELL
TST (SP)+ ;: CLEAN CHARACTER OFF OF STACK
BR 5$ ;: EXIT
3$: CMP (SP),#23 ;: IS IT A CONTROL-S?
BNE 32$ ;: BRANCH IF NO
CLR $TKS ;: DISABLE TTY KEYBOARD INTERRUPTS
TST (SP)+ ;: CLEAN CHAR OFF STACK
31$: TST $TKS ;: WAIT FOR A CHAR
BPL 31$ ;: LOOP UNTIL ITS THERE
MOV $TKB,-(SP) ;: GET THE CHARACTER
BIC #↑C177,(SP) ;: MAKE IT 7-BIT ASCII
CMP (SP)+,#21 ;: IS IT A CONTROL-Q?
BNE 31$ ;: BRANCH IF NO
MOV #100,$TKS ;: REENABLE TTY KEYBOARD INTERRUPTS
RTI ;: RETURN
32$: INC $TKCNT ;: COUNT THIS CHARACTER
CMP (SP),#140 ;: IS IT UPPER CASE?
BLT 4$ ;: BRANCH IF YES
CMP (SP),#175 ;: IS IT A SPECIAL CHAR?

```



```

9369 042352 003002          BGT      4$          ;; BRANCH IF YES
9370 042354 042716 000040    BIC      #40,(SP)    ;; MAKE IT UPPER CASE
9371 042360 112677 177530    4$:     MOVB     (SP)+,@$TKQIN ;; AND PUT IT IN QUEUE
9372 042364 005237 042114    INC      $TKQIN     ;; UPDATE THE POINTER
9373 042370 023727 042114 042121    CMP      $TKQIN,$$TKQEND ;; GO OFF THE END?
9374 042376 001003          BNE      5$          ;; BRANCH IF NO
9375 042400 012737 042120 042114    MOV      $$TKQSR,$TKQIN ;; RESET THE POINTER
9376 042406 000002          5$:     RTI           ;; RETURN
9377
9378 ;; *****
9379 ;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
9380 ;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
9381 ;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
9382 ;; *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
9383 042410 022737 000176 001140    $CKSWR: CMP      $$SWREG,$SWR    ;; IS THE SOFT-SWR SELECTED
9384 042416 001124          BNE      15$          ;; EXIT IF NOT
9385 042420 105777 136520    TSTB     @$TKS        ;; IS A CHAR WAITING?
9386 042424 100121          BPL      15$          ;; IF NOT, EXIT
9387 042426 117746 136514    MOVB     @$TKB,-(SP)   ;; YES
9388 042432 042716 177600    BIC      #1C177,(SP)  ;; MAKE IT 7-BIT ASCII
9389 042436 021627 000007    CMP      (SP),#7      ;; IS IT A CONTROL-G?
9390 042442 001300          BNE      2$          ;; IF NOT, PUT IT IN THE TTY QUEUE
9391 ;; AND EXIT
9392
9393 ;; *****
9394 ;; *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
9395 ;; *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
9396 ;; *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
9397 042444 123727 001134 000001    6$:     CMPB     $AUTOB,#1    ;; ARE WE RUNNING IN AUTO-MODE?
9398 042452 001674          BEQ      2$          ;; BRANCH IF YES
9399 042454 005726          TST      (SP)+        ;; CLEAR CONTROL-G OFF STACK
9400 042456 004737 042122    JSR      PC,$TKINT    ;; FLUSH THE TTY INPUT QUEUE
9401 042462 005077 136456    CLR      @$TKS        ;; DISABLE TTY KEYBOARD INTERRUPTS
9402 042466 112737 000001 001135    MOVB     #1,$INTAG    ;; SET INTERRUPT MODE INDICATOR
9403
9404 042474 104401 043320    $GTSWR: TYPE     ,$SCNTLG    ;; ECHO THE CONTROL-G (1G)
9405 042500 104401 043325    TYPE     ,$MSWR       ;; TYPE CURRENT CONTENTS
9406 042504 013746 000176    MOV      $SWREG,-(SP) ;; SAVE SWREG FOR TYPEOUT
9407 042510 104402          TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9408 042512 104401 043336    TYPE     ,$MNEW       ;; PROMPT FOR NEW SWR
9409 042516 005046          19$:    CLR      -(SP)       ;; CLEAR COUNTER
9410 042520 005046          CLR      -(SP)       ;; THE NEW SWR
9411 042522 105777 136416    7$:     TSTB     @$TKS        ;; CHAR THERE?
9412 042526 100375          BPL      7$          ;; IF NOT TRY AGAIN
9413
9414 042530 117746 136412    MOVB     @$TKB,-(SP)  ;; PICK UP CHAR
9415 042534 042716 177600    BIC      #1C177,(SP) ;; MAKE IT 7-BIT ASCII
9416
9417 042540 021627 000003          CMP      (SP),#3      ;; IS IT A CONTROL-C?
9418 042544 001015          BNE      9$          ;; BRANCH IF NOT
9419 042546 104401 043306    TYPE     ,$SCNTLC     ;; YES, ECHO CONTROL-C (1C)
9420 042552 062706 000006          ADD      #6,SP        ;; CLEAN UP STACK
9421 042556 123727 001135 000001    CMPB     $INTAG,#1    ;; REENABLE TTY KEYBOARD INTERRUPTS?
9422 042564 001003          BNE      8$          ;; BRANCH IF NO
9423 042566 012777 000100 136350    MOV      #100,@$TKS   ;; ALLOW TTY KEYBOARD INTERRUPTS
9424 042574 000137 001736    8$:     JMP      START    ;; CONTROL-C RESTART
    
```

```

9460 042600 021627 000025 99: CMP (SP),#25 :: IS IT A CONTROL-U?
9461 042604 001005 BNE 10$ :: BRANCH IF NOT
9462 042606 104401 043313 TYPE ,SCNTLU :: YES, ECHO CONTROL-U (↑U)
9463 042612 062706 000006 20$: ADD #6,SP :: IGNORE PREVIOUS INPUT
9464 042616 000737 BR 19$ :: LET'S TRY IT AGAIN

9465 042620 021627 000015 10$: CMP (SP),#15 :: IS IT A <CR>?
9466 042624 001022 BNE 16$ :: BRANCH IF NO
9467 042626 005766 000004 TST 4(SP) :: YES, IS IT THE FIRST CHAR?
9468 042632 001403 BEQ 11$ :: BRANCH IF YES
9469 042634 016677 000002 136276 MOV 2(SP),@SWR :: SAVE NEW SWR
9470 042642 062706 000006 11$: ADD #6,SP :: CLEAR UP STACK
9471 042646 104401 001273 14$: TYPE ,SCRLF :: ECHO <CR> AND <LF>
9472 042652 123727 001135 000001 CMPB $INTAG,#1 :: RE-ENABLE TTY KBD INTERRUPTS?
9473 042660 001003 BNE 15$ :: BRANCH IF NOT
9474 042662 012777 000100 136254 MOV #100,@STKS :: RE-ENABLE TTY KBD INTERRUPTS
9475 042670 000002 15$: RTI :: RETURN
9476 042672 004737 041370 16$: JSR PC,$TYPEC :: ECHO CHAR
9477 042676 021627 000060 CMP (SP),#60 :: CHAR < 0?
9478 042702 002420 BLT 18$ :: BRANCH IF YES
9479 042704 021627 000067 CMP (SP),#67 :: CHAR > ??
9480 042710 003015 BGT 18$ :: BRANCH IF YES
9481 042712 042726 000060 BIC #60,(SP)+ :: STRIP-OFF ASCII
9482 042716 005766 000002 TST 2(SP) :: IS THIS THE FIRST CHAR
9483 042722 001403 BEQ 17$ :: BRANCH IF YES
9484 042724 006316 ASL (SP) :: NO, SHIFT PRESENT
9485 042726 006316 ASL (SP) :: CHAR OVER TO MAKE
9486 042730 006316 ASL (SP) :: ROOM FOR NEW ONE.
9487 042732 005266 000002 17$: INC 2(SP) :: KEEP COUNT OF CHAR
9488 042736 056616 177776 BIS -2(SP),(SP) :: SET IN NEW CHAR
9489 042742 000667 BR 7$ :: GET THE NEXT ONE
9490 042744 104401 001272 18$: TYPE ,SQUES :: TYPE ?<CR><LF>
9491 042750 000720 BR 20$ :: SIMULATE CONTROL-U
9492 .DSABL LSB

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

9493 $RDCHR: MOV (SP),-(SP) :: PUSH DOWN THE PC AND
9494 042752 011646 000004 000002 MOV 4(SP),2(SP) :: THE PS
9495 042754 016666 000004 CLR 4(SP) :: GET READY FOR A CHARACTER
9496 042762 005066 000004 CLR -(SP) :: PUT NEW PS ON STACK
9497 042766 005046 042776 MOV #64$,-(SP) :: PUT NEW PC ON STACK
9498 042770 012746 042776 RTI :: POP NEW PC AND PS
9499 042774 000002 64$:
9500 042776 042776 64$: TST $TKCNT :: WAIT ON A CHARACTER
9501 042776 005737 042112 1$:
9502 042776 001775 BEQ 1$

```

```

9481 043004 005337 042112          DEC   $TKCNT          ;; DECREMENT THE COUNTER
9482 043010 117766 177102 000004      MOVB  $STKQOUT,4(SP) ;; GET ONE CHARACTER
9483 043016 005237 042116          INC   $TKQOUT        ;; UPDATE THE POINTER
9484 043022 023727 042116 042121  CMP   $TKQOUT,$STKGEND ;; DID IT GO OFF OF THE END?
9485 043030 001003          BNE   2$            ;; BRANCH IF NO
9486 043032 012737 042120 042116  MOV   $STKQRT,$STKQOUT ;; RESET THE POINTER
9487 043040 000002          RTI                    ;; RETURN
9488                                     *****
9489                                     *THIS ROUTINE WILL INPUT A STRING FROM THE TTY
9490                                     *CALL:
9491                                     *
9492                                     *   RDLIN          ;; INPUT A STRING FROM THE TTY
9493                                     *   RETURN HERE    ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
9494                                     *                                     ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
9495
9495 043042 010346          $RDLIN: MOV   R3,-(SP)          ;; SAVE R3
9496 043044 005046          CLR   -(SP)            ;; CLEAR THE RUBOUT KEY
9497 043046 012703 043276      1$:  MOV   $TTYIN,R3        ;; GET ADDRESS
9498 043052 022703 043306      2$:  CMP   $TTYIN+8.,R3     ;; BUFFER FULL?
9499 043056 101456          BLOS  4$              ;; BR IF YES
9500 043060 104410          RDCHR          ;; GO READ ONE CHARACTER FROM THE TTY
9501 043062 112613          MOVB  (SP)+,(R3)       ;; GET CHARACTER
9502 043064 122713 000177      10$: CMPB  #177,(R3)        ;; IS IT A RUBOUT
9503 043070 001022          BNE   5$              ;; BR IF NO
9504 043072 005716          TST   (SP)            ;; IS THIS THE FIRST RUBOUT?
9505 043074 001007          BNE   6$              ;; BR IF NO
9506 043076 112737 000134 043274  MOVB  #' \,9$          ;; TYPE A BACK SLASH
9507 043104 104401 043274          TYPE  ,9$
9508 043110 012716 177777          MOV   #-1,(SP)        ;; SET THE RUBOUT KEY
9509 043114 005303          6$:  DEC   R3            ;; BACKUP BY ONE
9510 043116 020327 043276      CMP   R3,$TTYIN       ;; STACK EMPTY?
9511 043122 103434          BLO   4$              ;; BR IF YES
9512 043124 111337 043274      MOVB  (R3),9$         ;; SETUP TO TYPEOUT THE DELETED CHAR.
9513 043130 104401 043274          TYPE  ,9$
9514 043134 000746          BR    2$              ;; GO TYPE
9515 043136 005716          5$:  TST   (SP)            ;; GO READ ANOTHER CHAR.
9516 043140 001406          BEQ   7$              ;; RUBOUT KEY SET?
9517 043142 112737 000134 043274  MOVB  #' \,9$          ;; BR IF NO
9518 043150 104401 043274          TYPE  ,9$            ;; TYPE A BACK SLASH
9519 043154 005016          CLR   (SP)            ;; CLEAR THE RUBOUT KEY
9520 043156 122713 000025      7$:  CMPB  #25,(R3)        ;; IS CHARACTER A CTRL U?
9521 043162 001003          BNE   8$              ;; BR IF NO
9522 043164 104401 043313          TYPE  ,SCNTLU        ;; TYPE A CONTROL "U"
9523 043170 000726          BR    1$              ;; GO START OVER
9524 043172 122713 000022      8$:  CMPB  #22,(R3)        ;; IS CHARACTER A "↑R"?
9525 043176 001011          BNE   3$              ;; BRANCH IF NO
9526 043200 105013          CLRB  (R3)            ;; CLEAR THE CHARACTER
9527 043202 104401 001273          TYPE  ,SCRLF         ;; TYPE A "CR" & "LF"
9528 043206 104401 043276          TYPE  ,TTYIN         ;; TYPE THE INPUT STRING
9529 043212 000717          BR    2$              ;; GO PICKUP ANOTHER CHARACTER
9530 043214 104401 001272      4$:  TYPE  ,QUES          ;; TYPE A '?'
9531 043220 000712          BR    1$              ;; CLEAR THE BUFFER AND LOOP
9532 043222 111337 043274      3$:  MOVB  (R3),9$         ;; ECHO THE CHARACTER
9533 043226 104401 043274          TYPE  ,9$
9534 043232 122723 000015          CMPB  #15,(R3)+      ;; CHECK FOR RETURN
9535 043236 001305          BNE   2$              ;; LOOP IF NOT RETURN
9536 043240 105063 177777          CLRB  -1(R3)         ;; CLEAR RETURN (THE 15)
    
```

```

9537 043244 104401 001274          TYPE      SLF          ;; TYPE A LINE FEED
9538 043250 005726          TST      (SP)+        ;; CLEAN RUBOUT KEY FROM THE STACK
9539 043252 012603          MOV      (SP)+,R3     ;; RESTORE R3
9540 043254 011646          MOV      (SP),-(SP)   ;; ADJUST THE STACK AND PUT ADDRESS OF THE
9541 043256 016666 000004 000002  MOV      4(SP),2(SP)  ;; FIRST ASCII CHARACTER ON IT
9542 043264 012766 043276 000004  MOV      #STTYIN,4(SP)
9543 043272 000002          RTI                    ;; RETURN
9544 043274          000          9$:      .BYTE      0          ;; STORAGE FOR ASCII CHAR. TO TYPE
9545 043275          000          .BYTE      0          ;; TERMINATOR
9546 043276 000010          STTYIN:  .BLKB      8.      ;; RESERVE 8 BYTES FOR TTY INPUT
9547 043306 041536 005015 000      SCNTLC:  .ASCIZ    /!C/<15><12>  ;; CONTROL "C"
9548 043313          136 006525 000012  SCNTLU:  .ASCIZ    /!U/<15><12>  ;; CONTROL "U"
9549 043320 043536 005015 000      SCNTLG:  .ASCIZ    /!G/<15><12>  ;; CONTROL "G"
9550 043325          015 051412 051127  SMSWR:   .ASCIZ    <15><12>/SWR = /
9551 043332 036440 000040          9551:  .ASCIZ    / NEW = /
9552 043336 020040 042516 020127  SMNEW:   .ASCIZ    / NEW = /
9553 043344 020075          000
9554          043350
9555          .EVEN
9556          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
9557
9558          ;; *****
9559          ;; *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9560          ;; *CHANGE IT TO BINARY.
9561          ;; *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9562          ;; *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
9563          ;; *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9564          ;; *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9565          ;; *CALL:
9566          ;; *   RDOCT          ;; READ AN OCTAL NUMBER
9567          ;; *   RETURN HERE    ;; LOW ORDER BITS ARE ON TOP OF THE STACK
9568          ;; *                   ;; HIGH ORDER BITS ARE IN $HIOCT
9569 043350 011646          SRDOCT:  MOV      (SP),-(SP)  ;; PROVIDE SPACE FOR THE
9570 043352 016666 000004 000002  MOV      4(SP),2(SP)  ;; INPUT NUMBER
9571 043360 010046          MOV      R0,-(SP)    ;; PUSH R0 ON STACK
9572 043362 010146          MOV      R1,-(SP)    ;; PUSH R1 ON STACK
9573 043364 010246          MOV      R2,-(SP)    ;; PUSH R2 ON STACK
9574 043366 104411          1$:      RDLIN          ;; READ AN ASCII LINE
9575 043370 012600          MOV      (SP)+,R0    ;; GET ADDRESS OF 1ST CHARACTER
9576 043372 010037 043476          MOV      R0,$$      ;; AND SAVE IT
9577 043376 005001          CLR      R1          ;; CLEAR DATA WORD
9578 043400 005002          CLR      R2
9579 043402 112046          2$:      MOVVB   (R0)+,-(SP)  ;; PICKUP THIS CHARACTER
9580 043404 001420          BEQ      3$          ;; IF ZERO GET OUT
9581 043406 122716 000060          CMPB    #'0,(SP)    ;; MAKE SURE THIS CHARACTER
9582 043412 003026          BGT      4$          ;; IS AN OCTAL DIGIT
9583 043414 122716 000067          CMPB    #'7,(SP)
9584 043420 002423          BLT      4$
9585 043422 006301          ASL     R1          ;; *2
9586 043424 006102          ROL     R2
9587 043426 006301          ASL     R1          ;; *4
9588 043430 006102          ROL     R2
9589 043432 006301          ASL     R1          ;; *8
9590 043434 006102          ROL     R2
9591 043436 042716 177770          BIC     #!C7,(SP)   ;; STRIP THE ASCII JUNK
9592 043442 062601          ADD     (SP)+,R1    ;; ADD IN THIS DIGIT

```

E01

```

9593 043444 000756          BR      2$          ;; LOOP
9594 043446 005726          3$:  TST      (SP)+      ;; CLEAN TERMINATOR FROM STACK
9595 043450 010166 000012  MOV      R1,12(SP)    ;; SAVE THE RESULT
9596 043454 010237 043506  MOV      R2,$SHIOCT
9597 043460 012602          MOV      (SP)+,R2     ;; POP STACK INTO R2
9598 043462 012601          MOV      (SP)+,R1     ;; POP STACK INTO R1
9599 043464 012600          MOV      (SP)+,R0     ;; POP STACK INTO R0
9600 043466 000002          RTI
9601 043470 005726          4$:  TST      (SP)+      ;; RETURN
9602 043472 105010          CLRB     (R0)        ;; CLEAN PARTIAL FROM STACK
9603 043474 104401          TYPE
9604 043476 000000          ;; SET A TERMINATOR
9605 043500 104401 001272  5$:  .WORD    0        ;; TYPE UP THRU THE BAD CHAR.
9606 043504 000730          TYPE     $QUES      ;; "?" "CR" & "LF"
9607 043506 000000          BR      1$          ;; TRY AGAIN
9608          $SHIOCT: .WORD    0        ;; HIGH ORDER BITS GO HERE
9609          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
9610          ;;*****
9611          ;;*SAVE R0-R5
9612          ;;*CALL:
9613          ;;*   SAVREG
9614          ;;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
9615          ;;*
9616          ;;*TOP---(+16)
9617          ;;* +2---(+18)
9618          ;;* +4---R5
9619          ;;* +6---R4
9620          ;;* +8---R3
9621          ;;*+10---R2
9622          ;;*+12---R1
9623          ;;*+14---R0
9624
9625 043510          $SAVREG:
9626 043510 010046          MOV      R0,-(SP)    ;; PUSH R0 ON STACK
9627 043512 010146          MOV      R1,-(SP)    ;; PUSH R1 ON STACK
9628 043514 010246          MOV      R2,-(SP)    ;; PUSH R2 ON STACK
9629 043516 010346          MOV      R3,-(SP)    ;; PUSH R3 ON STACK
9630 043520 010446          MOV      R4,-(SP)    ;; PUSH R4 ON STACK
9631 043522 010546          MOV      R5,-(SP)    ;; PUSH R5 ON STACK
9632 043524 016646 000022  MOV      22(SP),-(SP) ;; SAVE PS OF MAIN FLOW
9633 043530 016646 000022  MOV      22(SP),-(SP) ;; SAVE PC OF MAIN FLOW
9634 043534 016646 000022  MOV      22(SP),-(SP) ;; SAVE PS OF CALL
9635 043540 016646 000022  MOV      22(SP),-(SP) ;; SAVE PC OF CALL
9636 043544 000002          RTI
9637
9638          ;;*RESTORE R0-R5
9639          ;;*CALL:
9640          ;;*   RESREG
9641 043546          $RESREG:
9642 043546 012666 000022  MOV      (SP)+,22(SP) ;; RESTORE PC OF CALL
9643 043552 012666 000022  MOV      (SP)+,22(SP) ;; RESTORE PS OF CALL
9644 043556 012666 000022  MOV      (SP)+,22(SP) ;; RESTORE PC OF MAIN FLOW
9645 043562 012666 000022  MOV      (SP)+,22(SP) ;; RESTORE PS OF MAIN FLOW
9646 043566 012605          MOV      (SP)+,R5     ;; POP STACK INTO R5
9647 043570 012604          MOV      (SP)+,R4     ;; POP STACK INTO R4
9648 043572 012603          MOV      (SP)+,R3     ;; POP STACK INTO R3

```

# F01

```

9649 043574 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
9650 043576 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
9651 043600 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
9652 043602 000002      RTI
9653                      .SBTTL  POWER DOWN AND UP ROUTINES
9654
9655                      ;:*****
9656                      ;:POWER DOWN ROUTINE
9657 043604 012737 043744 000024 $PWRDN: MOV      #SILLUP,@#PWRVEC ;;SET FOR FAST UP
9658 043612 012737 000340 000026      MOV      #340,@#PWRVEC+2 ;;PRIO:7
9659 043620 010046      MOV      RD,-(SP)      ;;PUSH RD ON STACK
9660 043622 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9661 043624 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
9662 043626 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
9663 043630 010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
9664 043632 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
9665 043634 017746 135300      MOV      @SWR,-(SP)     ;;PUSH @SWR ON STACK
9666 043640 010637 043750      MOV      SP,$SAVR6     ;;SAVE SP
9667 043644 012737 043656 000024      MOV      #SPWRUP,@#PWRVEC ;;SET UP VECTOR
9668 043652 000000      HALT
9669 043654 000776      BR      .-2           ;;HANG UP
9670
9671                      ;:*****
9672                      ;:POWER UP ROUTINE
9673 043656 012737 043744 000024 $PWRUP: MOV      #SILLUP,@#PWRVEC ;;SET FOR FAST DOWN
9674 043664 013706 043750      MOV      $SAVR6,SP     ;;GET SP
9675 043670 005037 043750      CLR      $SAVR6       ;;WAIT LOOP FOR THE TTY
9676 043674 005237 043750      1$: INC      $SAVR6    ;;WAIT FOR THE INC
9677 043700 001375      BNE      1$           ;;OF WORD
9678 043702 012677 135232      MOV      (SP)+,@SWR   ;;POP STACK INTO @SWR
9679 043706 012605      MOV      (SP)+,R5     ;;POP STACK INTO R5
9680 043710 012604      MOV      (SP)+,R4     ;;POP STACK INTO R4
9681 043712 012603      MOV      (SP)+,R3     ;;POP STACK INTO R3
9682 043714 012602      MOV      (SP)+,R2     ;;POP STACK INTO R2
9683 043716 012601      MOV      (SP)+,R1     ;;POP STACK INTO R1
9684 043720 012600      MOV      (SP)+,R0     ;;POP STACK INTO R0
9685 043722 012737 043604 000024      MOV      #SPWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
9686 043730 012737 000340 000026      MOV      #340,@#PWRVEC+2 ;;PRIO:7
9687 043736 104401      TYPE      SPOWER      ;;REPORT THE POWER FAILURE
9688 043740 043752      SPWRMG: .WORD      SPOWER ;;POWER FAIL MESSAGE POINTER
9689 043742 000002      RTI
9690 043744 000000      $ILLUP: HALT          ;;THE POWER UP SEQUENCE WAS STARTED
9691 043746 000776      BR      .-2           ;;BEFORE THE POWER DOWN WAS COMPLETE
9692 043750 000000      $SAVR6: 0             ;;PUT THE SP HERE
9693 043752 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
9694 043760 000122
9695                      .EVEN
9696                      .SBTTL  TRAP DECODER
9697
9698                      ;:*****
9699                      ;:*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
9700                      ;:*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
9701                      ;:*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
9702                      ;:*GO TO THAT ROUTINE.
9703
9704 043762 010046      $TRAP: MOV      RD,-(SP) ;;SAVE RD

```

```

9705 043764 016600 000002      MOV      2(SP),RO      ;;GET TRAP ADDRESS
9706 043770 005740             TST      -(RO)        ;;BACKUP BY 2
9707 043772 111000             MOVVB   (RO),RO       ;;GET RIGHT BYTE OF TRAP
9708 043774 006300             ASL     RO            ;;POSITION FOR INDEXING
9709 043776 016000 044016      MOV     $TRPAD(RO),RO  ;;INDEX TO TABLE
9710 044002 000200             RTS     RO            ;;GO TO ROUTINE
  
```

;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

```

9715 044004 011646             $STRAP2: MOV      (SP),-(SP)  ;;MOVE THE PC DOWN
9716 044006 016666 000004 000002  MOV     4(SP),2(SP)  ;;MOVE THE PSW DOWN
9717 044014 000002             RTI                        ;;RESTORE THE PSW
  
```

.SBTTL TRAP TABLE

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

			ROUTINE		
9725			-----		
9726	044016	044004	\$STRPAD: .WORD	\$STRAP2	TRAP+1(104401) TTY TYPEOUT ROUTINE
9727	044020	041156	\$TYPE	;;CALL=TYPE	TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
9728	044022	041710	\$TYPOC	;;CALL=TYPOC	TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
9729	044024	041664	\$TYPOS	;;CALL=TYPOS	TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
9730	044026	041724	\$TYPON	;;CALL=TYPON	TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
9731	044030	041440	\$TYPDS	;;CALL=TYPDS	
9732					
9733	044032	042500	\$GTSWR	;;CALL=GTSWR	TRAP+6(104406) GET SOFT-SWR SETTING
9734					
9735	044034	042410	\$CKSWR	;;CALL=CKSWR	TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
9736	044036	042752	\$RDCHR	;;CALL=RDCHR	TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
9737	044040	043042	\$RDLIN	;;CALL=RDLIN	TRAP+11(104411) TTY TYPEIN STRING ROUTINE
9738	044042	043350	\$RDOCT	;;CALL=RDOCT	TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
9739	044044	043510	\$SAVREG	;;CALL=SAVREG	TRAP+13(104413) SAVE RO-R5 ROUTINE
9740	044046	043546	\$RESREG	;;CALL=RESREG	TRAP+14(104414) RESTORE RO-R5 ROUTINE
9741	044050	032652	\$SCOPI\$	;;CALL=SCOPI	TRAP+15(104415) INTERNAL LOOP ON ERROR
9742	044052	034452	\$SSINIT	;;CALL=TSSINIT	TRAP+16(104416) INITIALIZE SUBSYSTEM
9743	044054	033364	\$LOADRK	;;CALL=TLOADRK	TRAP+17(104417) LOAD RK611 FOR OPERATION
9744	044056	033446	\$GETRK	;;CALL=TGETRK	TRAP+20(104420) GET RK611 REGISTERS
9745	044060	035436	\$CHKOP	;;CALL=TCHKOP	TRAP+21(104421) CHECK OPREATION FOR ANY ERRORS
9746	044062	035404	\$CHKWE	;;CALL=TCHKWE	TRAP+22(104422) CHECK OPERATION FOR EXPECTED ERRORS
9747	044064	033160	\$IWAT16	;;CALL=TWAT16	TRAP+23(104423) WAIT 16 MS
9748	044066	033150	\$IWAT32	;;CALL=TWAT32	TRAP+24(104424) WAIT 32 MS
9749	044070	033140	\$IWAT48	;;CALL=TWAT48	TRAP+25(104425) WAIT 48 MS
9750	044072	033130	\$IWAT64	;;CALL=TWAT64	TRAP+26(104426) WAIT 64 MS
9751	044074	033120	\$IWAT80	;;CALL=TWAT80	TRAP+27(104427) WAIT 80 MS
9752	044076	033110	\$IWAT96	;;CALL=TWAT96	TRAP+30(104430) WAIT 96 MS
9753	044100	033100	\$IWAT112	;;CALL=TWAT112	TRAP+31(104431) WAIT 112 MS
9754	044102	033070	\$IWAT128	;;CALL=TWAT128	TRAP+32(104432) WAIT 128 MS
9755	044104	033060	\$IWAT144	;;CALL=TWAT144	TRAP+33(104433) WAIT 144 MS
9756	044106	033050	\$IWAT159	;;CALL=TWAT159	TRAP+34(104434) WAIT 160 MS
9757	044110	033040	\$IWAT1\$	;;CALL=TWAT1\$	TRAP+35(104435) WAIT FOR 1 SECOND
9758	044112	033030	\$IWAT2\$	;;CALL=TWAT2\$	TRAP+36(104436) WAIT FOR 2 SECONDS
9759	044114	033010	\$IWAT8\$	;;CALL=TWAT8\$	TRAP+37(104437) WAIT FOR 8 SECONDS
9760	044116	033020	\$IWAT1M	;;CALL=TWAT1M	TRAP+40(104440) WAIT FOR 1 MIN

H01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(732) 01-OCT-76 10:50 PAGE 181  
DZR6KB.P11 TRAP TABLE

SEQ 0214

9761

000102

STERM=-\$TRPAD



```

9762          .SBTTL DATA PATTERNS
9763          ;DATA PATTERN 1
9764          ;          PATTERN IS ALL ZEROS
9765
9766          ;DATA PATTERN 2
9767          ;          HI-LO FREQ. MIX
9768          PAT02:
9769          044120 177777
9770          044122 177777
9771          044124 177777
9772          044126 052525
9773          044130 052525
9774          044132 052525
9775          044134 177777
9776          044136 177777
9777          044140 052525
9778          044142 052525
9779          044144 177777
9780          044146 052525
9781          044150 177252
9782          044152 177252
9783          044154 172765
9784          044156 172765
9785
9786          ;DATA PATERN 3
9787          ;          HI FREQ. PHASE MIX
9788          PAT03:
9789          044160 000000
9790          044162 000000
9791          044164 000000
9792          044166 177777
9793          044170 177777
9794          044172 177777
9795          044174 000000
9796          044176 000000
9797          044200 177777
9798          044202 177777
9799          044204 000000
9800          044206 177777
9801          044210 000000
9802          044212 177777
9803          044214 000000
9804          044216 177777
9805
9806          ;DATA PATTERN 4
9807          ;          LO FREQ. PHASE MIX
9808          PAT04:
9809          044220 052525
9810          044222 052525
9811          044224 052525
9812          044226 125252
9813          044230 125252
9814          044232 125252
9815          044234 052525
9816          044236 052525
9817          044240 125252

```

9818	044242	125252	125252
9819	044244	052525	052525
9820	044246	125252	125252
9821	044250	052525	052525
9822	044252	125252	125252
9823	044254	052525	052525
9824	044256	125252	125252

;DATA PATTERN 5  
 ; MAX PRECOMP. PHASE MIX

9825			
9826			
9827			
9828	044260		
9829	044260	133333	133333
9830	044262	066666	066666
9831	044264	155555	155555
9832	044266	155555	155555
9833	044270	133333	133333
9834	044272	066666	066666
9835	044274	066666	066666
9836	044276	155555	155555
9837	044300	155555	155555
9838	044302	133333	133333
9839	044304	133333	133333
9840	044306	133333	133333
9841	044310	133333	133333
9842	044312	133333	133333
9843	044314	133333	133333
9844	044316	133333	133333

;DATA PATTERN 6  
 ; ROTATING BOUNDARY PULSE PRECOMP.

9845			
9846			
9847			
9848	044320		
9849	044320	121105	121105
9850	044322	150442	150442
9851	044324	064221	064221
9852	044326	132110	132110
9853	044330	055044	055044
9854	044332	026422	026422
9855	044334	013211	013211
9856	044336	105504	105504
9857	044340	042642	042642
9858	044342	021321	021321
9859	044344	110550	110550
9860	044346	044264	044264
9861	044350	022132	022132
9862	044352	011055	011055
9863	044354	104426	104426
9864	044356	042213	042213

;DATA PATTERN 7  
 ; FIELD OF ALL ONES

;DATA PATTERN 10  
 ; ROTATING CELL PULSE PRECOMP.

9865			
9866			
9867			
9868			
9869			
9870			
9871	044360		
9872	044360	026455	026455
9873	044362	113226	113226

9874	044364	045513	045513
9875	044366	122645	122645
9876	044370	151322	151322
9877	044372	064551	064551
9878	044374	132264	132264
9879	044376	055132	055132
9880	044400	026455	026455
9881	044402	113226	113226
9882	044404	045513	045513
9883	044406	122645	122645
9884	044410	151322	151322
9885	044412	064551	064551
9886	044414	132264	132264
9887	044416	055132	055132
9888			
9889			

;DATA PATTERN 11  
SHIFTED 1 IN A FIELD OF ZEROS

PAT11:

9891	044420	000001	000001
9892	044420	000002	000002
9893	044422	000002	000002
9894	044424	000004	000004
9895	044426	000010	000010
9896	044430	000020	000020
9897	044432	000040	000040
9898	044434	000100	000100
9899	044436	000200	000200
9900	044440	000400	000400
9901	044442	001000	001000
9902	044444	002000	002000
9903	044446	004000	004000
9904	044450	010000	010000
9905	044452	020000	020000
9906	044454	040000	040000
9907	044456	100000	100000

;DATA PATTERN 12  
SHIFTED 0 IN A FIELD OF ONES

PAT12:

9911	044460		
9912	044460	177776	177776
9913	044462	177775	177775
9914	044464	177773	177773
9915	044466	177767	177767
9916	044470	177757	177757
9917	044472	177737	177737
9918	044474	177677	177677
9919	044476	177577	177577
9920	044500	177377	177377
9921	044502	176777	176777
9922	044504	175777	175777
9923	044506	173777	173777
9924	044510	167777	167777
9925	044512	157777	157777
9926	044514	137777	137777
9927	044516	077777	077777

;DATA PATTERN 13

9888  
9889  
9890  
9891  
9892  
9893  
9894  
9895  
9896  
9897  
9898  
9899  
9900  
9901  
9902  
9903  
9904  
9905  
9906  
9907  
9908  
9909  
9910  
9911  
9912  
9913  
9914  
9915  
9916  
9917  
9918  
9919  
9920  
9921  
9922  
9923  
9924  
9925  
9926  
9927  
9928  
9929

9930			;PAT13: ALTERNATING 0-1
9931	044520		
9932	044520	052525	052525
9933	044522	052525	052525
9934	044524	052525	052525
9935	044526	052525	052525
9936	044530	052525	052525
9937	044532	052525	052525
9938	044534	052525	052525
9939	044536	052525	052525
9940	044540	052525	052525
9941	044542	052525	052525
9942	044544	052525	052525
9943	044546	052525	052525
9944	044550	052525	052525
9945	044552	052525	052525
9946	044554	052525	052525
9947	044556	052525	052525

9948			
9949			;DATA PATTERN 14
9950			ALTERNATING 1-0
9951	044560		;PAT14:
9952	044560	125252	125252
9953	044562	125252	125252
9954	044564	125252	125252
9955	044566	125252	125252
9956	044570	125252	125252
9957	044572	125252	125252
9958	044574	125252	125252
9959	044576	125252	125252
9960	044600	125252	125252
9961	044602	125252	125252
9962	044604	125252	125252
9963	044606	125252	125252
9964	044610	125252	125252
9965	044612	125252	125252
9966	044614	125252	125252
9967	044616	125252	125252

9968			
9969			;DATA PATTERN 15
9970			SHIFTING ZEROS AND ONES
9971	044620		;PAT15:
9972	044620	000001	000001
9973	044622	000003	000003
9974	044624	000007	000007
9975	044626	000017	000017
9976	044630	000037	000037
9977	044632	000077	000077
9978	044634	000177	000177
9979	044636	000377	000377
9980	044640	000777	000777
9981	044642	001777	001777
9982	044644	003777	003777
9983	044646	007777	007777
9984	044650	017777	017777
9985	044652	037777	037777

9986	044654	077777	077777
9987	044656	177777	177777
9988			
9989			
9990			
9991	044660		
9992	044660	072307	072307
9993	044662	135143	135143
9994	044664	156461	156461
9995	044666	167230	167230
9996	044670	073514	073514
9997	044672	035646	035646
9998	044674	016723	016723
9999	044676	107351	107351
10000	044700	143564	143564
10001	044702	061672	061672
10002	044704	030735	030735
10003	044706	114356	114356
10004	044710	046167	046167
10005	044712	123073	123073
10006	044714	151453	151453
10007	044716	164616	164616
10008			

:DATA PATTERN 16  
: PAT16: COMPOSITE ROTATING

```

10009          .SBTTL FIELDS AND VARIABLES FOR OPERATION CHECKING
10010          024000 CS1ERBIT=24000          ;CS1 ERROR BITS SPAR & CTO
10011          177400 CS2ERBIT=177400        ;CS2 ERROR BITS
10012          ;DLT,WCE,UPE,NED,NEM
10013          ;PGE,MOS,UFE
10014          000070 DSERBIT=70           ;DS ERROR BITS
10015          ;SPDLSS,DROT,ACLO
10016
10017 044720 000000 EXPWC: .WORD 0          ;EXPECTED WORD COUNT (GIVEN)
10018 044722 000000 EXPUBA: .WORD 0        ;EXPECTED UPPER BA (COMPUTED)
10019 044724 000000 EXPBA: .WORD 0        ;EXPECTED BUS ADDRESS (COMPUTED)
10020 044726 000000 EXPCYL: .WORD 0       ;EXPECTED CYLINDER (COMPUTED)
10021 044730 000000 EXPSEC: .WORD 0       ;EXPECTED SECTOR (COMPUTED)
10022 044732 000000 EXPTRK: .WORD 0      ;EXPECTED TRACK (COMPUTED)
10023
10024 044734 000000 REALWC: .WORD 0       ;WORD COUNT AT END OF OPERATION
10025 044736 000000 REALUB: .WORD 0      ;REAL UPPER BA
10026 044740 000000 REALBA: .WORD 0      ;BUS ADDRESS
10027 044742 000000 REALCY: .WORD 0      ;CYLINDER
10028 044744 000000 REALTRK: .WORD 0     ;TRACK
10029 044746 000000 REALSEC: .WORD 0    ;SECTOR
10030
10031 044750 000000 GRP1ER: .WORD 0          ;GROUP 1 ERROR FIELDS
10032          000001 SPARERR=BIT0        ;CONTROLLER DETECTED DRIVE BUS PARITY ERR
10033          000002 SKIERR= BIT1        ;SEEK INCOMPLETE
10034          000004 NXFERR= BIT2        ;NON-EXECUTABLE DRIVE FUNCTION
10035          000010 DRPARERR=BIT3      ;DRIVE DETECTED DRIVE BUS PARITY ERROR
10036          000020 FMTERR= BIT4        ;FORMAT ERROR
10037          000040 DTYERR= BITS       ;DRIVE TYPE ERROR
10038          000100 ACLOERR=BIT6       ;AC LOW ERROR
10039          000200 SPDERR= BIT7        ;SPEED LOSS ERROR
10040          000400 DROTERR=BIT8       ;DRIVE OFF TRACK ERROR
10041          001000 COERR= BIT9         ;CYLINDER OVER FLOW ERROR
10042          002000 IDAERR= BIT10      ;ILLEGAL DISK ADDRESS ERROR
10043          004000 WLERR= BIT11       ;WRITE LOCK ERROR
10044          010000 DTERR= BIT12      ;DRIVE TIMING ERROR
10045          020000 NCERWE= BIT13     ;NO CERR WITH ERROR SET ERROR
10046          040000 UNSERR= BIT14     ;DRIVE UNSAFE ERROR
10047          100000 CERNER= BIT15     ;CERR BUT NO ERROR SET ERROR
10048
10049 044752 000000 GRP2ER: .WORD 0          ;GROUP 2 ERROR FIELD
10050          000001 ECHERR= BIT0       ;ECC HARD ERROR
10051          000002 DCKERR= BIT1       ;DATA CHECK ERROR
10052          000004 WCKERR= BIT2       ;WRITE CHECK ERROR
10053          000010 DLTERR= BIT3       ;DATA LATE ERROR
10054          000020 OPIERR= BIT4       ;OPERATION INCOMPLETE ERROR
10055          000040 HVCERR=BITS       ;HEADER VRC ERROR
10056          000100 BSERR= BIT6        ;BAD SECTOR ERROR
10057
10058 044754 000000 GRP3ER: .WORD 0          ;GROUP 3 ERROR FIELD
10059          000001 NEDERR= BIT0       ;NON-EXISTANT DRIVE ERROR
10060          000002 CTOERR= BIT1       ;CONTROLLER TIME OUT ERROR
10061          000004 UFERR= BIT2        ;UNIT FIELD ERROR
10062          000010 MDSERR= BIT3       ;MULTIPLE DRIVE SELECT ERROR
10063          000020 PGERR= BIT4        ;PROGRAMMING ERROR
10064          000040 NEMERR= BITS       ;NON-EXISTANT MEMORY ERROR
    
```

10065		000100	UPERR= BIT6	:UNIBUS PARITY ERROR
10066		000200	ILFERR= BIT7	:ILLEGAL FUNCTION ERROR.
10067				
10068	044756	000000	GRP4ER: .WORD 0	:GROUP 4 ERROR FIELD
10069		000001	WCERR= BIT0	:WORD COUNT ERROR FLAG
10070		000002	UBAERR= BIT1	:UPPER BA ERROR
10071		000004	BAERR= BIT2	:BUS ADDRESS ERROR FLAG
10072		000010	CYLERR= BIT3	:CYL ADDRESS ERROR FLAG
10073		000020	TRKERR= BIT4	:TRACK ADDRESS ERROR FLAG
10074		000040	SECERR= BITS	:SECTOR ADDRESS ERROR FLAG
10075				
10076	044760	050550	GRP4MS: .WORD EM10	
10077	044762	050623	.WORD EM11A	
10078	044764	050575	.WORD EM11	
10079	044766	050677	.WORD EM12	
10080	044770	050732	.WORD EM13	
10081	044772	050762	.WORD EM14	
10082				
10083	044774	051014	GRP3MS: .WORD EM15	
10084	044776	051037	.WORD EM16	
10085	045000	051062	.WORD EM17	
10086	045002	051103	.WORD EM18	
10087	045004	051130	.WORD EM19	
10088	045006	051152	.WORD EM20	
10089	045010	051176	.WORD EM21	
10090	045012	051222	.WORD EM22	
10091				
10092	045014	051243	GRP2MS: .WORD EM23	
10093	045016	051254	.WORD EM24	
10094	045020	051267	.WORD EM25	
10095	045022	051303	.WORD EM26	
10096	045024	051315	.WORD EM27	
10097	045026	051342	.WORD EM28	
10098	045030	051355	.WORD EM29	
10099				
10100	045032	051376	GRP1MS: .WORD EM30	
10101	045034	051451	.WORD EM31	
10102	045036	051471	.WORD EM32	
10103	045040	051527	.WORD EM33	
10104	045042	051575	.WORD EM34	
10105	045044	051612	.WORD EM35	
10106	045046	051633	.WORD EM36	
10107	045050	051642	.WORD EM37	
10108	045052	051665	.WORD EM38	
10109	045054	051705	.WORD EM39	
10110	045056	051727	.WORD EM40	
10111	045060	051761	.WORD EM41	
10112	045062	052002	.WORD EM42	
10113	045064	052025	.WORD EM43	
10114	045066	052067	.WORD EM44	
10115	045070	052104	.WORD EM45	
10116				
10117	045072	000000	GPSUMF: .WORD 0	:GROUP ERROR SUMMARY FLAGS
10118		000001	GRP1ST= BIT0	:GROUP 1 ERROR SET
10119		000002	GRP2ST= BIT1	:GROUP 2 ERROR SET
10120		000004	GRP3ST= BIT2	:GROUP 3 ERROR SET

10121 000010 GP1NR= BIT3 ;GROUP 1 ERROR NOT RECEIVED  
10122 000020 GP2NR= BIT4 ;GROUP 2 ERROR NOT RECEIVED  
10123 000040 GP3NR= BIT5 ;GROUP 3 ERROR NOT RECEIVED  
10124 040000 DRSTER= BIT14 ;ERROR IN GETTING DRIVE STATUS FLAG.  
10125 100000 REPNR= BIT15 ;REPORTING NOT RECEIVED SWITCH

10127 .SBTTL TABLE OF OPERATION MESSAGE ADDRESS  
10128 ;\* THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE  
10129 ;\* USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.  
10130

10131	045074	045134	CMNDLB:	.WORD	OPER00	:ADDRESS OF SELECT MESSAGE
10132	045076	045151		.WORD	OPER02	PACK ACK
10133	045100	045162		.WORD	OPER04	DRIVE CLEAR
10134	045102	045176		.WORD	OPER06	UNLOAD
10135	045104	045205		.WORD	OPER10	START SPINDLE
10136	045106	045223		.WORD	OPER12	RECALIBRATE
10137	045110	045237		.WORD	OPER14	OFFSET
10138	045112	045246		.WORD	OPER16	SEEK
10139	045114	045253		.WORD	OPER20	READ DATA
10140	045116	045265		.WORD	OPER22	WRITE DATA
10141	045120	045300		.WORD	OPER24	READ HEADER
10142	045122	045315		.WORD	OPER26	WRITE HEADER
10143	045124	045333		.WORD	OPER30	WRITE CHECK
10144	045126	045347		.WORD	OPER32	ILLEGAL OPERATION 33
10145	045130	045373		.WORD	OPER34	35
10146	045132	045417		.WORD	OPER36	37
10147						

10148 .SBTTL OPERATION MESSAGES  
10149 045134 051104 053111 020105 OPER00: .ASCIZ /DRIVE SELECT/  
10150 045142 042523 042514 052103  
10151 045150 000  
10152 045151 120 041501 020113 OPER02: .ASCIZ /PACK ACK/  
10153 045156 041501 000113  
10154 045162 051104 053111 020105 OPER04: .ASCIZ /DRIVE CLEAR/  
10155 045170 046103 040505 000122  
10156 045176 047125 047514 042101 OPER06: .ASCIZ /UNLOAD/  
10157 045204 000  
10158 045205 123 040524 052122 OPER10: .ASCIZ /START SPINDLE/  
10159 045212 051440 044520 042116  
10160 045220 042514 000  
10161 045223 122 041505 046101 OPER12: .ASCIZ /RECALIBRATE/  
10162 045230 041111 040522 042524  
10163 045236 000  
10164 045237 117 043106 042523 OPER14: .ASCIZ /OFFSET/  
10165 045244 000124  
10166 045246 042523 045505 000 OPER16: .ASCIZ /SEEK/  
10167 045253 122 040505 020104 OPER20: .ASCIZ /READ DATA/  
10168 045260 040504 040524 000  
10169 045265 127 044522 042524 OPER22: .ASCIZ /WRITE DATA/  
10170 045272 042040 052101 000101  
10171 045300 042522 042101 044040 OPER24: .ASCIZ /READ HEADERS/  
10172 045306 040505 042504 051522  
10173 045314 000  
10174 045315 127 044522 042524 OPER26: .ASCIZ /WRITE HEADERS/  
10175 045322 044040 040505 042504  
10176 045330 051522 000



10177	045333	127	044522	042524	OPER30: .ASCIZ /WRITE CHECK/
10178	045340	041440	042510	045503	
10179	045346	000			
10180	045347	111	046114	043505	OPER32: .ASCIZ /ILLEGAL FUNCTION 33/
10181	045354	046101	043040	047125	
10182	045362	052103	047511	020116	
10183	045370	031463	000		
10184	045373	111	046114	043505	OPER34: .ASCIZ /ILLEGAL FUNCTION 35/
10185	045400	046101	043040	047125	
10186	045406	052103	047511	020116	
10187	045414	032463	000		
10188	045417	111	046114	043505	OPER36: .ASCIZ /ILLEGAL FUNCTION 37/
10189	045424	046101	043040	047125	
10190	045432	052103	047511	020116	
10191	045440	033463	000		
10192	045443	127	044522	042524	OPER37: .ASCIZ /WRITE DATA ABORTED WITH BSE/
10193	045450	042040	052101	020101	
10194	045456	041101	051117	042524	
10195	045464	020104	044527	044124	
10196	045472	041040	042523	000	
10197	045477				OPER40:
10198	045477	127	044522	042524	OPER41: .ASCIZ /WRITE CHECK ABORTED WITH WCE/
10199	045504	041440	042510	045503	
10200	045512	040440	047502	052122	
10201	045520	042105	053440	052111	
10202	045526	020110	041527	000105	
10203					

```

10204
10205
10206 045534 020040 000
10207 045537 015 051012 033113
10208 045544 030461 041040 051525
10209 045552 040440 042104 042522
10210 045560 051523 024040 000040
10211 045566 024440 036440 000040
10212 045574 045522 030466 020061
10213 045602 042526 052103 051117
10214 045610 040440 042104 042522
10215 045616 051523 024040 000040
10216 045624 045522 030466 020061
10217 045632 051120 047511 044522
10218 045640 054524 024040 000040
10219 045646 047111 052523 043106
10220 045654 041511 042511 052116
10221 045662 046440 046505 051117
10222 045670 027131 050040 047522
10223 045676 051107 046501 040440
10224 045704 047502 052122 047111
10225 045712 027107 005015 000
10226 045717 015 052012 020117
10227 045724 054502 040520 051523
10228 045732 052040 051505 044524
10229 045740 043516 042040 044522
10230 045746 042526 030040 020054
10231 045754 046120 041501 020105
10232 045762 052111 047440 043106
10233 045770 046055 047111 105
10234 045775 015 052012 020117
10235 046002 042524 052123 042040
10236 046010 044522 042526 030040
10237 046016 020054 042522 046120
10238 046024 041501 020105 051120
10239 046032 043517 040522 020115
10240 046040 040520 045503 053440
10241 046046 052111 020110 041523
10242 046054 040522 041524 020110
10243 046062 040520 045503 000
10244 046067 015 042012 044522
10245 046074 042526 030040 053440
10246 046102 046111 020114 047516
10247 046110 020124 042502 052040
10248 046116 051505 042524 027104
10249 046124 051440 040524 052122
10250 046132 040440 020124 030462
10251 046140 020064 047524 052040
10252 046146 051505 020124 051104
10253 046154 030440 000
10254 046157 015 047012 020117
10255 046164 051104 053111 051505
10256 046172 040440 040526 046111
10257 046200 041101 042514 043040
10258 046206 051117 052040 051505
10259 046214 044524 043516 020056

```

.SBTTL ASCII MESSAGES

```

SPACE2: .ASCIZ / /
OPR001: .ASCIZ <15><12>/RK611 BUS ADDRESS ( /

```

```

OPR002: .ASCIZ / ) = /
OPR003: .ASCIZ /RK611 VECTOR ADDRESS ( /

```

```

OPR004: .ASCIZ /RK611 PRIORITY ( /

```

```

OPR005: .ASCIZ /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>

```

```

OPR006: .ASCII <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/

```

```

.ASCIZ <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/

```

```

OPR007: .ASCIZ <15><12>/DRIVE 0 WILL NOT BE TESTED. START AT 214 TO TEST DR 1/

```

```

OPR008: .ASCIZ <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/

```

10260 046222 051120 043517 040522  
10261 046230 020115 041101 051117  
10262 046236 042524 000104  
10263 046242 005015 044124 020105  
10264 046250 047506 046114 053517  
10265 046256 047111 020107 051104  
10266 046264 053111 051505 053440  
10267 046272 046111 020114 042502  
10268 046300 052040 051505 042524  
10269 046306 006504 000012  
10270 046312 005015 047516 050040  
10271 046320 051101 052111 020131  
10272 046326 050117 044524 047117  
10273 046334 052455 044516 052502  
10274 046342 020123 040520 044522  
10275 046350 054524 042440 051122  
10276 046356 051117 040440 042116  
10277 046364 005015  
10278 046366 032062 034050 020051  
10279 046374 042523 052103 051117  
10280 046402 043040 051117 040515  
10281 046410 020124 040504 040524  
10282 046416 054040 042506 020122  
10283 046424 042524 052123 041040  
10284 046432 050131 051501 042523  
10285 046440 006504 000012  
10286 046444 005015 042515 047515  
10287 046452 054522 051440 055111  
10288 046460 020105 047516 020124  
10289 046466 040514 043522 020105  
10290 046474 047105 052517 044107  
10291 046502 043040 051117 041040  
10292 046510 051525 040440 042104  
10293 046516 042522 051523 041040  
10294 046524 052111 020123 033061  
10295 046532 023040 030440 020067  
10296 046540 042524 052123 006523  
10297 046546 012  
10298 046547 101 046114 042040  
10299 046554 052101 020101 043130  
10300 046562 051105 052040 051505  
10301 046570 051524 053440 052111  
10302 046576 020110 042101 051104  
10303 046604 037040 000  
10304 046607 063 045462 000  
10305 046613 066 045464 000  
10306 046617 071 045466 000  
10307 046623 040 054502 040520  
10308 046630 051523 042105 005015  
10309 046636 000  
10310 046637 015 005012 020040  
10311 046644 020040 020040 025040  
10312 046652 025052 041440 052501  
10313 046660 044524 047117 025040  
10314 046666 025052 005015 012  
10315 046673 124 044510 020123

OPR009: .ASCIZ <15><12>/THE FOLLOWING DRIVES WILL BE TESTED/<15><12>

OPR010: .ASCII <15><12>/NO PARITY OPTION-UNIBUS PARITY ERROR AND/<15><12>

.ASCIZ /24(8) SECTOR FORMAT DATA XFER TEST BYPASSED/<15><12>

OPR011: .ASCII <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS

.ASCIZ /ALL DATA XFER TESTS WITH ADDR >/

OPR012: .ASCIZ /32K/  
OPR013: .ASCIZ /64K/  
OPR014: .ASCIZ /96K/  
OPR015: .ASCIZ / BYPASSED/<15><12>

OPR016: .ASCII <15><12><12>/ \*\*\* CAUTION \*\*\*/<15><12><12>

.ASCII /THIS PROGRAM SHOULD BE HALTED ONLY AT END OF PASS./<15><12>

10316 046700 051120 043517 040522  
 10317 046706 020115 044123 052517  
 10318 046714 042114 041040 020105  
 10319 046722 040510 052114 042105  
 10320 046730 047440 046116 020131  
 10321 046736 052101 042440 042116  
 10322 046744 047440 020106 040520  
 10323 046752 051523 006456 012  
 10324 046757 111 020106 040510  
 10325 046764 052114 042105 040440  
 10326 046772 020124 047101 020131  
 10327 047000 052117 042510 020122  
 10328 047006 044524 042515 020054  
 10329 047014 044124 020105 052123  
 10330 047022 052101 020105 043117  
 10331 047030 052040 042510 042040  
 10332 047036 044522 042526 005015  
 10333 047044 051117 041440 051101  
 10334 047052 051124 042111 042507  
 10335 047060 041440 047101 047516  
 10336 047066 020124 042502 050040  
 10337 047074 042522 044504 052103  
 10338 047102 042105 006456 005012  
 10339 047110 046101 020114 051104  
 10340 047116 053111 051505 052040  
 10341 047124 020117 042502 052040  
 10342 047132 051505 042524 020104  
 10343 047140 052515 052123 041040  
 10344 047146 020105 047117 046055  
 10345 047154 047111 026105 005015  
 10346 047162 042522 042101 026131  
 10347 047170 040440 042116 053440  
 10348 047176 044522 042524 046040  
 10349 047204 041517 020113 042522  
 10350 047212 042523 027124 005015  
 10351 047220 047101 020131 051104  
 10352 047226 053111 020105 047516  
 10353 047234 020124 047524 041040  
 10354 047242 020105 042524 052123  
 10355 047250 042105 046440 051525  
 10356 047256 020124 042502 047440  
 10357 047264 043106 046055 047111  
 10358 047272 027105 005015 012  
 10359 047277 116 052117 035105  
 10360 047304 031040 042116 040440  
 10361 047312 042116 051440 041125  
 10362 047320 042523 052521 047105  
 10363 047326 020124 040520 051523  
 10364 047334 051040 047125 052040  
 10365 047342 046511 020105 051511  
 10366 047350 005015  
 10367 047352 020040 020040 020040  
 10368 047360 050101 051120 054117  
 10369 047366 033440 046440 047111  
 10370 047374 052125 051505 043040  
 10371 047402 051117 042440 041501

.ASCII /IF HALTED AT ANY OTHER TIME, THE STATE OF THE DRIVE/<15><12>

.ASCII /OR CARTRIDGE CANNOT BE PREDICTED./<15><12><12>

.ASCII /ALL DRIVES TO BE TESTED MUST BE ON-LINE,/<15><12>

.ASCII /READY, AND WRITE LOCK RESET./<15><12>

.ASCII /ANY DRIVE NOT TO BE TESTED MUST BE OFF-LINE./<15><12><12>

.ASCII /NOTE: 2ND AND SUBSEQUENT PASS RUN TIME IS/<15><12>

.ASCIZ / APPROX 7 MINUTES FOR EACH DRIVE./<15><12>

10372 047410 020110 051104 053111  
10373 047416 027105 005015 000  
10374 047423 015 043012 051111  
10375 047430 052123 031040 033065  
10376 047436 051440 041505 047524  
10377 047444 051522 047040 052117  
10378 047452 041040 042523 042440  
10379 047460 051122 051117 043040  
10380 047466 042522 027105  
10381 047472 040515 044530 052515  
10382 047500 020115 040504 040524  
10383 047506 052040 047522 051516  
10384 047514 042506 020122 042524  
10385 047522 052123 041040 050131  
10386 047530 051501 042523 006504  
10387 047536 000012  
10398 047540 020040 020040 006440  
10389 047546 047412 046116 020131  
10390 047554 020061 051104 053111  
10391 047562 027105 047440 042526  
10392 047570 046122 050101 042520  
10393 047576 020104 050117 051105  
10394 047604 052101 047511 051516  
10395 047612 041040 050131 051501  
10396 047620 042523 006504 000012  
10397 047626 005015 041523 050117  
10398 047634 035105 041440 030510  
10399 047642 024040 051124 043511  
10400 047650 026051 042440 031465  
10401 047656 034055 020073 044103  
10402 047664 026062 042440 034464  
10403 047672 031055 024040 041501  
10404 047700 041440 052517 046120  
10405 047706 026105 027040 053062  
10406 047714 041457 024515  
10407 047720 005015 042101 052512  
10408 047726 052123 051040 031067  
10409 047734 043040 051117 041440  
10410 047742 047117 052123 047101  
10411 047750 020124 042514 042526  
10412 047756 020114 047117 041440  
10413 047764 031110 005015 000  
10414 047771 015 050012 047522  
10415 047776 051107 046501 040440  
10416 050004 047502 052122 047111  
10417 050012 020107 042502 040503  
10418 050020 051525 020105 051105  
10419 050026 047522 020122 044124  
10420 050034 042522 044123 046117  
10421 050042 020104 054105 042503  
10422 050050 042105 042105 005015  
10423 050056 000  
10424  
10425  
10426 050057 106 052101 046101  
10427 050064 047055 047117 042440

OPRO17: .ASCII &lt;15&gt;&lt;12&gt;/FIRST 256 SECTORS NOT BSE ERROR FREE./

.ASCIZ /MAXIMUM DATA TRANSFER TEST BYPASSED/&lt;15&gt;&lt;12&gt;

OPRO18: .ASCIZ / /&lt;15&gt;&lt;12&gt;/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/&lt;15&gt;&lt;12&gt;

OPRO19: .ASCII &lt;15&gt;&lt;12&gt;@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@

.ASCIZ &lt;15&gt;&lt;12&gt;/ADJUST R72 FOR CONSTANT LEVEL ON CH2/&lt;15&gt;&lt;12&gt;

ABORT: .ASCIZ &lt;15&gt;&lt;12&gt;/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDED/&lt;15&gt;&lt;12&gt;

## .SBTTL ERROR MESSAGES

EM1: .ASCIZ /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/

10428	050072	044530	052123	047101	
10429	050100	020124	042515	047515	
10430	050106	054522	040440	020124	
10431	050114	045522	030466	020061	
10432	050122	040502	042523	040440	
10433	050130	042104	042522	051523	
10434	050136	000			
10435	050137	106	052101	046101	EM2: .ASCIZ /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/
10436	050144	053455	044522	042524	
10437	050152	051040	040505	054504	
10438	050160	040440	042116	044440	
10439	050166	020105	044504	020104	
10440	050174	047516	020124	040503	
10441	050202	051525	020105	047111	
10442	050210	042524	051122	050125	
10443	050216	000124			
10444	050220	040506	040524	026514	EM3: .ASCIZ /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
10445	050226	040520	044522	054524	
10446	050234	042440	051122	051117	
10447	050242	052040	040522	027120	
10448	050250	05C740	020103	052101	
10449	050256	042440	051122	051117	
10450	050264	036440	000040		
10451	050270	054105	042520	052103	EM4: .ASCIZ /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS: /
10452	050276	042105	044440	052116	
10453	050304	051105	052522	052120	
10454	050312	042040	042111	047040	
10455	050320	052117	047440	041503	
10456	050326	051125	047440	020122	
10457	050334	040527	020123	040514	
10458	050342	042524	020056	047503	
10459	050350	046515	047101	020104	
10460	050356	040527	035123	000	
10461	050363	123	041125	054523	EM5: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET ERROR/
10462	050370	052123	046505	041440	
10463	050376	042514	051101	042040	
10464	050404	042111	047040	052117	
10465	050412	051040	051505	052105	
10466	050420	042440	051122	051117	
10467	050426	000			
10468	050427	123	041125	054523	EM6: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
10469	050434	052123	046505	041440	
10470	050442	042514	051101	042040	
10471	050450	042111	047040	052117	
10472	050456	051040	051505	052105	
10473	050464	042040	053105	041511	
10474	050472	020105	047111	042524	
10475	050500	051122	050125	000124	
10476	050506	052523	051502	051531	EM7: .ASCIZ /SUBSYSTEM CLEAR DID NOT SET READY/
10477	050514	042524	020115	046103	
10478	050522	040505	020122	044504	
10479	050530	020104	047516	020124	
10480	050536	042523	020124	042522	
10481	050544	042101	000131		
10482	050550	047527	042122	041440	EM10: .ASCIZ /WORD COUNT INCORRECT/
10483	050556	052517	052116	044440	

10484	050564	041516	051117	042522	
10485	050572	052103	000		
10486	050575	102	051525	040440	EM11: .ASCIZ /BUS ADDRESS INCORRECT/
10487	050602	042104	042522	051523	
10488	050610	044440	041516	051117	
10489	050616	042522	052103	000	
10490	050623	125	050120	051105	EM11A: .ASCIZ /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
10491	050630	041040	051525	040440	
10492	050636	042104	042522	051523	
10493	050644	041040	052111	020123	
10494	050652	047111	047503	051122	
10495	050660	041505	020124	041050	
10496	050666	030501	026066	030440	
10497	050674	024467	000		
10498	050677	103	046131	047111	EM12: .ASCIZ /CYLINDER ADDRESS INCORRECT/
10499	050704	042504	020122	042101	
10500	050712	051104	051505	020123	
10501	050720	047111	047503	051122	
10502	050726	041505	000124		
10503	050732	051124	041501	020113	EM13: .ASCIZ /TRACK ADDRESS INCORRECT/
10504	050740	042101	051104	051505	
10505	050746	020123	047111	047503	
10506	050754	051122	041505	000124	
10507	050762	042523	052103	051117	EM14: .ASCIZ /SECTOR ADDRESS INCORRECT./
10508	050770	040440	042104	042522	
10509	050776	051523	044440	041516	
10510	051004	051117	042522	052103	
10511	051012	000056			
10512	051014	047516	026516	054105	EM15: .ASCIZ /NON-EXISTANT DRIVE/
10513	051022	051511	040524	052116	
10514	051030	042040	044522	042526	
10515	051036	000			
10516	051037	103	047117	051124	EM16: .ASCIZ /CONTROLLER TIMEOUT/
10517	051044	046117	042514	020122	
10518	051052	044524	042515	052517	
10519	051060	000124			
10520	051062	047125	052111	043040	EM17: .ASCIZ /UNIT FIELD ERROR/
10521	051070	042511	042114	042440	
10522	051076	051122	051117	000	
10523	051103	115	046125	050111	EM18: .ASCIZ /MULIPLE DRIVE SELECT/
10524	051110	042514	042040	044522	
10525	051116	042526	051440	046105	
10526	051124	041505	000124		
10527	051130	051120	043517	040522	EM19: .ASCIZ /PROGRAMMING ERROR/
10528	051136	046515	047111	020107	
10529	051144	051105	047522	000122	
10530	051152	047516	026516	054105	EM20: .ASCIZ /NON-EXISTANT MEMORY/
10531	051160	051511	040524	052116	
10532	051166	046440	046505	051117	
10533	051174	000131			
10534	051176	047125	041111	051525	EM21: .ASCIZ /UNIBUS PARITY ERROR/
10535	051204	050040	051101	052111	
10536	051212	020131	051105	047522	
10537	051220	000122			
10538	051222	046111	042514	040507	EM22: .ASCIZ /ILLEGAL FUNCTION/
10539	051230	020114	052506	041516	

10540	051236	044524	047117	000				
10541	051243	105	041503	044040	EM23:	.ASCIZ	/ECC HARD/	
10542	051250	051101	000104					
10543	051254	040504	040524	041440	EM24:	.ASCIZ	/DATA CHECK/	
10544	051262	042510	045503	000				
10545	051267	127	044522	042524	EM25:	.ASCIZ	/WRITE CHECK/	
10546	051274	041440	042510	045503				
10547	051302	000						
10548	051303	104	052101	020101	EM26:	.ASCIZ	/DATA LATE/	
10549	051310	040514	042524	000				
10550	051315	117	042520	040522	EM27:	.ASCIZ	/OPERATION INCOMPLETE/	
10551	051322	044524	047117	044440				
10552	051330	041516	046517	046120				
10553	051336	052105	000105					
10554	051342	042510	042101	051105	EM28:	.ASCIZ	/HEADER VRC/	
10555	051350	053040	041522	000				
10556	051355	102	042101	051440	EM29:	.ASCIZ	/BAD SECTOR ERROR/	
10557	051362	041505	047524	020122				
10558	051370	051105	047522	000122				
10559	051376	047503	052116	047522	EM30:	.ASCIZ	/CONTROLLER DETECTED DRIVE BUS PARITY ERROR/	
10560	051404	046114	051105	042040				
10561	051412	052105	041505	042524				
10562	051420	020104	051104	053111				
10563	051426	020105	052502	020123				
10564	051434	040520	044522	054524				
10565	051442	042440	051122	051117				
10566	051450	000						
10567	051451	123	042505	020113	EM31:	.ASCIZ	/SEEK INCOMPLETE/	
10568	051456	047111	047503	050115				
10569	051464	042514	042524	000				
10570	051471	116	047117	042455	EM32:	.ASCIZ	/NON-EXECUTABLE DRIVE FUNCTION/	
10571	051476	042530	052503	040524				
10572	051504	046102	020105	051104				
10573	051512	053111	020105	052506				
10574	051520	041516	044524	047117				
10575	051526	000						
10576	051527	104	044522	042526	EM33:	.ASCIZ	/DRIVE DETECTED DRIVE BUS PARITY ERROR/	
10577	051534	042040	052105	041505				
10578	051542	042524	020104	051104				
10579	051550	053111	020105	052502				
10580	051556	020123	040520	044522				
10581	051564	054524	042440	051122				
10582	051572	051117	000					
10583	051575	106	051117	040515	EM34:	.ASCIZ	/FORMAT ERROR/	
10584	051602	020124	051105	047522				
10585	051610	000122						
10586	051612	051104	053111	020105	EM35:	.ASCIZ	/DRIVE TYPE ERROR/	
10587	051620	054524	042520	042440				
10588	051626	051122	051117	000				
10589	051633	101	020103	047514	EM36:	.ASCIZ	/AC LOW/	
10590	051640	000127						
10591	051642	050123	047111	046104	EM37:	.ASCIZ	/SPINDLE SPEED LOSS/	
10592	051650	020105	050123	042505				
10593	051656	020104	047514	051523				
10594	051664	000						
10595	051665	104	044522	042526	EM38:	.ASCIZ	/DRIVE OFF TRACK/	



10596	051672	047440	043106	052040		
10597	051700	040522	045503	000		
10598	051705	103	046131	047111	EM39:	.ASCIZ /CYLINDER OVERFLOW/
10599	051712	042504	020122	053117		
10600	051720	051105	046106	053517		
10601	051726	000				
10602	051727	111	046114	043505	EM40:	.ASCIZ /ILLEGAL DISK PACK ADDRESS/
10603	051734	046101	042040	051511		
10604	051742	020113	040520	045503		
10605	051750	040440	042104	042522		
10606	051756	051523	000			
10607	051761	127	044522	042524	EM41:	.ASCIZ /WRITE LOCK ERROR/
10608	051766	046040	041517	020113		
10609	051774	051105	047522	000122		
10610	052002	051104	053111	020105	EM42:	.ASCIZ /DRIVE TIMING ERROR/
10611	052010	044524	044515	043516		
10612	052016	042440	051122	051117		
10613	052024	000				
10614	052025	116	020117	042503	EM43:	.ASCIZ /NO CERR WITH SOME OTHER ERROR SET/
10615	052032	051122	053440	052111		
10616	052040	020110	047523	042515		
10617	052046	047440	044124	051105		
10618	052054	042440	051122	051117		
10619	052062	051440	052105	000		
10620	052067	104	044522	042526	EM44:	.ASCIZ /DRIVE UNSAFE/
10621	052074	052440	051516	043101		
10622	052102	000105				
10623	052104	042503	051122	051440	EM45:	.ASCIZ /CERR SET BUT NO OTHER ERROR SET/
10624	052112	052105	041040	052125		
10625	052120	047040	020117	052117		
10626	052126	042510	020122	051105		
10627	052134	047522	020122	042523		
10628	052142	000124				
10629						
10630	052144	053126	042040	042111	EM46:	.ASCIZ /VV DID NOT SET WITH PACK ACK/
10631	052152	047040	052117	051440		
10632	052160	052105	053440	052111		
10633	052166	020110	040520	045503		
10634	052174	040440	045503	000		
10635	052201	123	040524	052524	EM47:	.ASCIZ /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
10636	052206	020123	040526	044514		
10637	052214	020104	042523	020124		
10638	052222	047117	051440	046105		
10639	052230	041505	020124	047524		
10640	052236	047040	047117	042455		
10641	052244	044530	052123	047101		
10642	052252	020124	051104	053111		
10643	052260	000105				
10644	052262	052123	052101	051525	EM48:	.ASCIZ /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
10645	052270	053040	046101	042111		
10646	052276	051040	051505	052105		
10647	052304	047440	020116	042523		
10648	052312	042514	052103	052040		
10649	052320	020117	054105	051511		
10650	052326	040524	052116	042040		
10651	052334	044522	042526	000		

10652	052341	123	040524	052524	EM49:	.ASCIZ	/STATUS VALID NOT RESET ON DRIVE RELEASE/
10653	052346	020123	040526	044514			
10654	052354	020104	047516	020124			
10655	052362	042522	042523	020124			
10656	052370	047117	042040	044522			
10657	052376	042526	051040	046105			
10658	052404	040505	042523	000			
10659	052411	105	050130	041505	EM50:	.ASCIZ	/EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
10660	052416	042524	020104	047062			
10661	052424	020104	047111	042524			
10662	052432	051122	050125	020124			
10663	052440	044504	020104	047516			
10664	052446	020124	041517	052503			
10665	052454	020122	051117	053440			
10666	052462	051501	046040	052101			
10667	052470	027105	041440	046517			
10668	052476	040515	042116	053440			
10669	052504	051501	000072				
10670	052510	040503	047116	052117	EM51:	.ASCII	/CANNOT READ BAD SECTOR FILE/<15><12>
10671	052516	051040	040505	020104			
10672	052524	040502	020104	042523			
10673	052532	052103	051117	043040			
10674	052540	046111	006505	012			
10675	052545	124	051505	044524		.ASCIZ	/TESTING ABORTED ON THIS DRIVE/
10676	052552	043516	040440	047502			
10677	052560	052122	042105	047440			
10678	052566	020116	044124	051511			
10679	052574	042040	044522	042526			
10680	052602	000					
10681	052603	101	044514	047107	EM52:	.ASCIZ	/ALIGNMENT PACK ON THIS DRIVE. TESTING ABORTED/
10682	052610	042515	052116	050040			
10683	052616	041501	020113	047117			
10684	052624	052040	044510	020123			
10685	052632	051104	053111	027105			
10686	052640	052040	051505	044524			
10687	052646	043516	040440	047502			
10688	052654	052122	042105	000			
10689	052661	124	020117	040515	EM53:	.ASCIZ	/TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
10690	052666	054516	041040	042101			
10691	052674	051440	041505	047524			
10692	052702	051522	020056	042524			
10693	052710	052123	047111	020107			
10694	052716	041101	051117	042524			
10695	052724	020104	047117	052040			
10696	052732	044510	020123	051104			
10697	052740	053111	000105				
10698	052744	042510	042101	051105	EM54:	.ASCIZ	/HEADER WORD MISCOMPARE/
10699	052752	053440	051117	020104			
10700	052760	044515	041523	046517			
10701	052766	040520	042522	000			
10702	052773	104	052101	020101	EM55:	.ASCIZ	/DATA WORD MISCOMPARE/
10703	053000	047527	042122	046440			
10704	053006	051511	047503	050115			
10705	053014	051101	000105				
10706							
10707	053020	040503	047116	052117	EM56:	.ASCIZ	/CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/

10708	053026	043040	047111	020104	
10709	053034	042510	042101	051105	
10710	053042	030040	044440	020116	
10711	053050	042522	042101	040440	
10712	053056	042116	051440	051117	
10713	053064	020124	042510	042101	
10714	053072	051105	000123		
10715	053076	052123	052101	051525	EMSVAL: .ASCIZ /STATUS VALID/
10716	053104	053040	046101	042111	
10717	053112	000			
10718	053113	123	046105	041505	EMNZPR: .ASCIZ /SELECT WITH NON-ZERO STATUS PAIR/
10719	053120	020124	044527	044124	
10720	053126	047040	047117	055055	
10721	053134	051105	020117	052123	
10722	053142	052101	051525	050040	
10723	053150	044501	000122		
10724	053154	051127	052111	047111	EMWCS2: .ASCIZ /WRITING COMMAND STATUS REGISTER 2/
10725	053162	020107	047503	046515	
10726	053170	047101	020104	052123	
10727	053176	052101	051525	051040	
10728	053204	043505	051511	042524	
10729	053212	020122	000062		
10730		051612			EMDTPE= EM35 ;DRIVE TYPE ERROR
10731	053216	051104	053111	020105	EMDI: .ASCIZ /DRIVE INTERRUPT/
10732	053224	047111	042524	051122	
10733	053232	050125	000124		
10734		051527			EMDPAR= EM33 ;DRIVE DETECTED DRIVE BUS PARITY ERROR
10735	053236	051104	053111	020105	EMDSC: .ASCIZ /DRIVE STATUS CHANGE/
10736	053244	052123	052101	051525	
10737	053252	041440	040510	043516	
10738	053260	000105			
10739	053262	051104	053111	020105	EMDA: .ASCIZ /DRIVE ATTENTION/
10740	053270	052101	042524	052116	
10741	053276	047511	000116		
10742	053302	047503	052116	047522	EMCCLR: .ASCIZ /CONTROLLER CLEAR/
10743	053310	046114	051105	041440	
10744	053316	042514	051101	000	
10745		045134			EMSELD= OPER00 ;DRIVE SELECT
10746		045162			EMDCLR= OPER04 ;DRIVE CLEAR
10747		045223			EMRCAL= OPER12 ;RECALIBRATE
10748	053323	123	041505	047117	EM2INT: .ASCIZ /SECOND INTERRUPT/
10749	053330	020104	047111	042524	
10750	053336	051122	050125	000124	
10751	053344	042523	045505	052040	EMSKSF: .ASCIZ /SEEK TO SELF/
10752	053352	020117	042523	043114	
10753	053360	000			
10754		045246			EMSK= OPER16 ;SEEK
10755	053361	125	042516	050130	EMUXIT: .ASCIZ /UNEXPECTED INTERRUPT/
10756	053366	041505	042524	020104	
10757	053374	047111	042524	051122	
10758	053402	050125	000124		
10759	053406	047125	041111	051525	EMUR: .ASCIZ /UNIBUS RESET/
10760	053414	051040	051505	052105	
10761	053422	000			
10762	053423	122	051505	052105	EMRSET: .ASCIZ /RESET/
10763	053430	000			

10764		051303		
10765	053431	122	040505	044504
10766	053436	043516	042040	052101
10767	053444	020101	052502	043106
10768	053452	051105	000	
10769	053455	103	047117	051124
10770	053462	046117	042514	020122
10771	053470	051105	047522	000122
10772	053476	052523	051502	051531
10773	053504	042524	020115	046103
10774	053512	040505	000122	
10775	053516	052515	052114	050111
10776	053524	042514	044440	052116
10777	053532	051105	052522	052120
10778	053540	000123		
10779	053542	040502	020104	042523
10780	053550	052103	051117	020123
10781	053556	047117	050040	041501
10782	053564	020113	047111	040440
10783	053572	042522	051501	052440
10784	053600	042523	020104	054502
10785	053606	052040	051505	020124
10786	053614	041450	046131	031440
10787	053622	031061	034050	024451
10788	053630	005015		
10789	053632	042524	052123	047111
10790	053640	020107	04101	051117
10791	053646	042524	021104	047117
10792	053654	052040	044510	020123
10793	053662	051104	053111	000105
10794				
10795				
10796	053670	051524	020124	052516
10797	053676	020115	051105	020122
10798	053704	041520	020040	051104
10799	053712	053111	000105	
10800	053716	051524	020124	052516
10801	053724	020115	051105	020122
10802	053732	041520	020040	051104
10803	053740	053111	000105	
10804	053744	045522	051503	020061
10805	053752	020040	045522	051503
10806	053760	020062	020040	045522
10807	053766	051504	020040	020040
10808	053774	045522	051105	020040
10809	054002	020040	045522	051501
10810	054010	043117	020040	045522
10811	054016	041504	046131	020040
10812	054024	045522	040504	000
10813	054031	122	041113	020101
10814	054036	020040	051040	053513
10815	054044	000103		
10816	054046	030101	020060	020040
10817	054054	020040	030102	020060
10818	054062	020040	020040	030101
10819	054070	020061	020040	020040

EMDLT= EM26 :DATA LATE  
EMRDB: .ASCIZ /READING DATA BUFFER/

EMCERR: .ASCIZ /CONTROLLER ERROR/

EMSCLR: .ASCIZ /SUBSYSTEM CLEAR/

EMMI: .ASCIZ /MULTIPLE INTERRUPTS/

DRVABT: .ASCII /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(B))<15><12>

.ASCIZ /TESTING ABORTED ON THIS DRIVE/

.SBTTL DATA HEADERS FOR ERROR REPORTS  
DH001: .ASCIZ /TST NUM ERR PC DRIVE/

DH002A: .ASCIZ /TST NUM ERR PC DRIVE/

DH002B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA/

DH002C: .ASCIZ /RKBA RKWC/

DH002D: .ASCIZ /A00 B00 A01 B01 A02 B02 A03 B03/

10820	054076	030102	020061	020040
10821	054104	020040	030101	020062
10822	054112	020040	020040	030102
10823	054120	020062	020040	020040
10824	054126	030101	020063	020040
10825	054134	020040	030102	000063
10826	054142	045522	051503	020061
10827	054150	020040	045522	051503
10828	054156	020062	020040	045522
10829	054164	051504	020040	020040
10830	054172	045522	051105	020040
10831	054200	020040	045522	051501
10832	054206	043117	020040	045522
10833	054214	051115	000061	
10834	054220	044124	020105	041101
10835	054226	053117	020105	051101
10836	054234	020105	054105	042520
10837	054242	052103	042105	042440
10838	054250	051122	051117	020123
10839	054256	044124	052101	042040
10840	054264	042111	047040	052117
10841	054272	051440	052105	044440
10842	054300	020116	050117	051105
10843	054306	052101	047511	035116
10844	054314	000		
10845	054315	124	042510	040440
10846	054322	047502	042526	040440
10847	054330	042522	052440	042516
10848	054336	050130	041505	042524
10849	054344	020104	051105	047522
10850	054352	051522	051440	052105
10851	054360	044440	020116	050117
10852	054366	051105	052101	047511
10853	054374	035116	000	
10854	054377	124	042510	040440
10855	054404	047502	042526	040440
10856	054412	042522	042440	051122
10857	054420	051117	020123	042523
10858	054426	020124	047111	047440
10859	054434	042520	040522	044524
10860	054442	047117	000072	
10861	054446	047101	020131	044506
10862	054454	046105	020104	044527
10863	054462	044124	040440	046114
10864	054470	047440	042516	020123
10865	054476	052515	052123	041040
10866	054504	020105	047503	051516
10867	054512	042111	051105	042105
10868	054520	044440	053116	046101
10869	054526	042111	000	
10870	054531	105	051122	051117
10871	054536	040440	020124	047503
10872	054544	050115	042514	044524
10873	054552	047117	047440	020106
10874	054560	050117	051105	052101
10875	054566	047511	000116	

DH003B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKMR1/

DH005: .ASCIZ /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/

DH006: .ASCIZ /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/

DH007: .ASCIZ /THE ABOVE ARE ERRORS SET IN OPERATION:/

DH005A: .ASCIZ /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/

DH010: .ASCIZ /ERROR AT COMPLETION OF OPERATION/

10876	054572	054105	052120	020104	DH010A: .ASCIZ /EXPTD IS/
10877	054600	020040	051511	000	
10878	054605	122	040505	020104	DH010B: .ASCIZ /READ DATA WITH FORCED CYLINDER OVERFLOW/
10879	054612	040504	040524	053440	
10880	054620	052111	020110	047506	
10881	054626	041522	042105	041440	
10882	054634	046131	047111	042504	
10883	054642	020122	053117	051105	
10884	054650	046106	053517	000	
10885	054655	116	052117	051440	DH011: .ASCIZ /NOT SET AS A RESULT OF/
10886	054662	052105	040440	020123	
10887	054670	020101	042522	052523	
10888	054676	052114	047440	000106	
10889	054704	047516	020124	042522	DH012: .ASCIZ /NOT RESET AS A RESULT OF/
10890	054712	042523	020124	051501	
10891	054720	040440	051040	051505	
10892	054726	046125	020124	043117	
10893	054734	000			
10894	054735	123	052105	040440	DH013: .ASCIZ /SET AS A RESULT OF/
10895	054742	020123	020101	042522	
10896	054750	052523	052114	047440	
10897	054756	000106			
10898	054760	042522	042523	020124	DH014: .ASCIZ /RESET AS A RESULT OF/
10899	054766	051501	040440	051040	
10900	054774	051505	046125	020124	
10901	055002	043117	000		
10902	055005	107	047517	020104	DH015: .ASCIZ /GOOD BAD WORD NUM/
10903	055012	020040	041040	042101	
10904	055020	020040	020040	053440	
10905	055026	051117	020104	052516	
10906	055034	000115			
10907	055036	052523	051502	051531	DH016: .ASCIZ /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
10908	055044	042524	020115	046103	
10909	055052	040505	020122	047524	
10910	055060	051040	051505	052105	
10911	055066	051440	042505	020113	
10912	055074	044514	044515	020124	
10913	055102	046101	047514	044527	
10914	055110	043516	044040	040505	
10915	055116	051504	052040	020117	
10916	055124	042522	047514	042101	
10917	055132	000			
10918	055133	123	046105	041505	DH017: .ASCIZ /SELECT AFTER RECAL/
10919	055140	020124	043101	042524	
10920	055146	020122	042522	040503	
10921	055154	000114			
10922	055156	042522	040503	020114	DH018: .ASCIZ /RECAL COMPLETE ATTN AFTER SELECT/
10923	055164	047503	050115	042514	
10924	055172	042524	040440	052124	
10925	055200	020116	043101	042524	
10926	055206	020122	042523	042514	
10927	055214	052103	000		
10928					
10929					.SBTTL DATA TABLES FOR ERROR REPORTS
10930		055220			.EVEN
10931	055220	001302	001116	001626	DT003: .WORD \$TESTN, \$ERRPC, DRVNUM, T.CS1, T.CS2, T.DS, T.ER, T.ASOF, T.MR1

10932	055226	001540	001550	001552		
10933	055234	001554	001556	001566		
10934	055242	001302	001116	001626	DT001:	.WORD \$TESTN,\$ERRPC,DRVNUM
10935	055250				DT002:	
10936	055250				DT004:	
10937	055250				DT005:	
10938	055250				DT006:	
10939	055250				DT007:	
10940	055250	001302	001116	001626	DT010:	.WORD \$TESTN,\$ERRPC,DRVNUM
10941	055256	001540	001550	001552		.WORD T.CS1,T.CS2,T.DS,T.ER,T.ASOF
10942	055264	001554	001556			
10943	055270	001560	001546			.WORD T.DCYL,T.DA
10944	055274	001544	001542			.WORD T.BA,T.WC
10945	055300	001202	001204	001206		.WORD \$REG10,\$REG11,\$REG12,\$REG13,\$REG14,\$REG15,\$REG16,\$REG17
10946	055306	001210	001212	001214		
10947	055314	001216	001220			
10948	055320	001302	001116	001626	DT015:	.WORD \$TESTN,\$ERRPC,DRVNUM
10949	055326	001202	001204	001206	DT015A:	.WORD \$REG10,\$REG11,\$REG12
10950					.SBTTL	DATA FORMATS FOR ERROR REPORTS
10951	055334	000001			DF001:	.WORD 1
10952	055336	003	000			.BYTE 3,0
10953						
10954	055340	000004			DF002:	.WORD 4
10955	055342	000	000			.BYTE 0,0
10956	055344	053716				.WORD DH002A
10957	055346	003	000			.BYTE 3,0
10958	055350	053744				.WORD DH002B
10959	055352	007	000			.BYTE 7,0
10960	055354	054031				.WORD DH002C
10961	055356	002	000			.BYTE 2,0
10962						
10963						
10964	055360	000002			DF003:	.WORD 2
10965	055362	003	000			.BYTE 3,0
10966	055364	054142				.WORD DH003B
10967	055366	006	000			.BYTE 6,0
10968						
10969	055370	000006			DF004:	.WORD 6
10970	055372	000	000			.BYTE 0,0
10971	055374	000000			DF004A:	.WORD 0
10972	055376	000	000			.BYTE 0,0
10973	055400	053716				.WORD DH002A
10974	055402	003	000			.BYTE 3,0
10975	055404	053744				.WORD DH002B
10976	055406	007	000			.BYTE 7,0
10977	055410	054031				.WORD DH002C
10978	055412	002	000			.BYTE 2,0
10979	055414	054046				.WORD DH002D
10980	055416	010	000			.BYTE 10,0
10981						
10982	055420	000007			DF005:	.WORD 7
10983	055422	000	000			.BYTE 0,0
10984	055424	000000			DF005A:	.WORD 0
10985	055426	000	000			.BYTE 0,0
10986	055430	053716				.WORD DH002A
10987	055432	003	000			.BYTE 3,0

10988	055434	053744		.WORD	DH002B
10989	055436	007	000	.BYTE	7,0
10990	055440	054031		.WORD	DH002C
10991	055442	002	000	.BYTE	2,0
10992	055444	054046		.WORD	DH002D
10993	055446	010	000	.BYTE	10,0
10994	055450	054446		.WORD	DH005A
10995	055452	000	000	.BYTE	0,0
10996					
10997	055454	000005		DF006: .WORD	5
10998	055456	000	000	.BYTE	0,0
10999	055460	000000		DF006A: .WORD	0
11000	055462	000	000	.BYTE	0,0
11001	055464	053716		.WORD	DH002A
11002	055466	003	000	.BYTE	3,0
11003	055470	053744		.WORD	DH002B
11004	055472	007	000	.BYTE	7,0
11005	055474	054031		.WORD	DH002C
11006	055476	002	000	.BYTE	2,0
11007					
11008	055500	000004		DF007: .WORD	4
11009	055502	000	000	.BYTE	0,0
11010	055504	000000		DF007A: .WORD	0
11011	055506	000	000	.BYTE	0,0
11012	055510	053716		.WORD	DH002A
11013	055512	003	000	.BYTE	3,0
11014	055514	054142		.WORD	DH003B
11015	055516	006	000	.BYTE	6,0
11016					
11017	055520	000004		DF010: .WORD	4
11018	055522	000	000	.BYTE	0,0
11019	055524	000000		DF010A: .WORD	0
11020	055526	000	000	.BYTE	0,0
11021	055530	053716		.WORD	DH002A
11022	055532	003	000	.BYTE	3,0
11023	055534	054572		.WORD	DH010A
11024	055536	002	000	.BYTE	2,0
11025					
11026	055540	000004		DF011: .WORD	4
11027	055542	000	000	.BYTE	0,0
11028	055544	000000		DF011A: .WORD	0
11029	055546	000	000	.BYTE	0,0
11030	055550	053716		.WORD	DH002A
11031	055552	003	000	.BYTE	3,0
11032	055554	054142		.WORD	DH003B
11033	055556	000006	000000	.WORD	6,0
11034					
11035	055562	000002		DF015: .WORD	2
11036	055564	003	000	.BYTE	3,0
11037	055566	055005		.WORD	DH015
11038	055570	003	000	.BYTE	3,0
11039					
11040	055572	000001		DF016: .WORD	1
11041	055574	003	000	.BYTE	3,0
11042					
11043	055576	000052		BS26: .BLKW	52



G03

SEQ 0239

11044	055722	000052	BS24:	.BLKW	52
11045	056046	000200	PATCH:	.BLKW	200
11046	056446	001000	IBUFF:	.BLKW	1000
11047	060446	001000	OBUFF:	.BLKW	1000
11048		000001	.END		

ABASE = 177440	179#	501	542	
ABORT = 047771	7433	10414#		
ACDW1 = 000000	501	544		
ACDW2 = 000000	501			
ACLO = 000C10	274#	8250		
ACLOER= 000100	8252	10038#		
ACPUOP= 000000	501	516		
ADDW0 = 000000	501			
ADDW1 = 000000	501			
ADDW10= 000000	501			
ADDW11= 000000	501			
ADDW12= 000000	501			
ADDW13= 000000	501			
ADDW14= 000000	501			
ADDW15= 000000	501			
ADDW2 = 000000	501			
ADDW3 = 000000	501			
ADDW4 = 000000	501			
ADDW5 = 000000	501			
ADDW6 = 000000	501			
ADDW7 = 000000	501			
ADDW8 = 000000	501			
ADDW9 = 000000	501			
ADEVCT= 000000	501	507		
ADEVN = 000000	501	543		
ADJCLK 041062	1403	9047#		
AENV = 000000	501	512		
AENVN = 000000	501	513		
AFATAL= 000000	501	504		
AMADR1= 000000	501	529		
AMADR2= 000000	501	533		
AMADR3= 000000	501	536		
AMADR4= 000000	501	539		
AMAMS1= 000000	501	523		
AMAMS2= 000000	501	531		
AMAMS3= 000000	501	534		
AMAMS4= 000000	501	537		
AMSGAD= 000000	501	509		
AMSGLG= 000000	501	510		
AMSGTY= 000000	501	503		
AMTYP1= 000000	501	524		
AMTYP2= 000000	501	532		
AMTYP3= 000000	501	535		
AMTYP4= 000000	501	538		
APASS = 000000	501	506		
APRIOR= 000240	178#	501		
APTCSU= 000040	7286#	9106		
APTENV= 000001	7242	7284#	7341	9099
APTSIZ= 000200	996	7283#		
APTSP0= 000100	7244	7285#	9101	
ASWREG= 000000	501	514		
ATESTN= 000000	501	505		
AUNIT = 000000	501	508		
AUSWR = 000000	501	515		
AVECT1= 000210	177#	501	540	
AVECT2= 000000	501	541		









EM27	051315	10096	10550#															
EM28	051342	10097	10554#															
EM29	051355	10098	10556#															
EM3	050220	7465	10444#															
EM3N	001400	618#	1356*	1368*	1382*	1439*	7971*	7973*	7984*									
EM30	051376	10100	10559#															
EM31	051451	10101	10567#															
EM32	051471	10102	10570#															
EM33	051527	10103	10576#	10734														
EM34	051575	10104	10583#															
EM35	051612	10105	10586#	10730														
EM36	051633	10106	10589#															
EM37	051642	10107	10591#															
EM38	051665	10108	10595#															
EM39	051705	10109	10598#															
EM4	050270	7654	7829	10451#														
EM4N	001410	699#	1716*															
EM40	051727	10110	10602#															
EM41	051761	10111	10607#															
EM42	052002	10112	10610#															
EM43	052025	10113	10614#															
EM44	052067	10114	10620#															
EM45	052104	10115	10623#															
EM46	052144	1382	10630#															
EM47	052201	1356	10635#															
EM48	052262	1368	10644#															
EM49	052341	1439	10652#															
EM5	050363	7971	10461#															
EM5N	001420	713#																
EM50	052411	1891	10659#															
EM51	052510	2197	2207	2217	2224	10670#												
EM52	052603	2233	10681#															
EM53	052661	2246	10689#															
EM54	052744	8740	10698#															
EM55	052773	8743	10702#															
EM56	053020	7857	10707#															
EM6	050427	7973	10468#															
EM6N	001430	738#																
EM7	050506	7984	10476#															
EM7N	001440	764#																
ERRCNT	001632	892#	7428*	7431														
ERRLMT	001634	894#	2391	2880	2959	3028	3107	3176	3257	3328	3402	3473	3700	3788				
		3871	3956	4041	4100	4157	4215	4271	4327	4460	4534	4675	4799	4837				
		5468																
ERRVEC=	000004	130#	981	982*	993*	7019	7020	7024*	7029*	7041*	7046*	7055*	7065*	7066*				
		7102	7103*	7105*	7108*													
EXPBA	044724	8023*	8050*	10019#														
EXPCYL	044726	8109*	8123*	8128	10020#													
EXPSEC	044730	4769	8127*	10021#														
EXPTRK	044732	4759	8126*	10022#														
EXPUBA	044722	8049*	8057	10018#														
EXPWC	044720	7999*	8001	8224	10017#													
FMTE =	000020	256#																
FMTERR=	000020	10036#																
FPFMT =	000400	916#	5311															
GENCOM	037526	2318	2386	2394	2443	2517	2595	2646	2699	2759	2819	2848	2874	2883				





IWAT16	033160	7635#	9747											
IWAT25	033030	7602#	9758											
IWAT32	033150	7632#	9748											
IWAT48	033140	7629#	9749											
IWAT64	033130	7626#	9750											
IWAT85	033010	7596#	9759											
IWAT80	033120	7623#	9751											
IWAT96	033110	7620#	9752											
KIPAR0=	172340	168#	7005	7033										
KIPAR1=	172342	169#												
KIPAR2=	172344	170#												
KIPAR3=	172346	171#												
KIPAR4=	172350	172#												
KIPAR5=	172352	173#												
KIPAR6=	172354	174#	8887*											
KIPAR7=	172356	175#	7049											
KIPDR0=	172300	157#												
KIPDR1=	172302	158#												
KIPDR2=	172304	159#												
KIPDR3=	172306	160#												
KIPDR4=	172310	161#												
KIPDR5=	172312	162#												
KIPDR6=	172314	163#												
KIPDR7=	172316	164#												
KWLADD	001674	929#	7476*	7514*	7802*	7848*								
KWLVEC	001676	930#	7511											
LCKHLR	032420	7475#	7512											
LCLKPR=	100000	917#	7521	7548	7645	7800	7846							
LCLKTK	001660	919#	1999*	2001	7475*	7547*	7554	7556*	7581*	7648	7818*	7824		
LF =	000012	41#	9149	9155										
LOADRK	033364	7698#	9743											
LRLOAD	033324	2175	2310	2338	2434	2462	2509	2542	2587	2607	2638	2659	2691	2719
		2751	2777	2811	2840	2859	2917	2941	2992	3009	3071	3087	3138	3155
		3215	3235	3290	3307	3352	3371	3427	3445	3496	3524	3568	3641	3664
		3679	3725	3748	3767	3811	3839	3853	3896	3920	3935	3981	4005	4020
		4079	4136	4194	4250	4306	4350	4377	4498	4513	4589	4639	4654	4708
		4737	4778	4813	4895	4936	4955	4997	5012	5032	5130	5145	5167	5256
		5273	5317	5351	5394	5417	5445	5518	5549	5571	5586	5610	5630	5645
		5683	5704	5719	5731	5775	5794	5808	5822	5876	5918	5935	5976	5991
		6003	6035	6067	6085	6123	6139	6154	6178	6193	6239	6276	6298	6313
		6328	6340	6371	6394	6432	6450	7678#	9052					
L.ASOF	001612	879#	6161*											
L.BA	001604	874#	4418*	4868*	6865*	8023	8038							
L.CS1	001600	872#	1233*	1272*	1340*	1371*	1420*	1460*	1510*	1553*	1554*	1597*	1644*	1663*
		1698#	1704*	1732*	1775*	1823*	1873*	1909*	1939*	1981*	2042*	2061*	2103*	2122*
		4414#	4442*	4866*	6533*	6588*	6659*	6671*	6813*	6823*	6834*	6842*	6855*	6866*
		7681	7713	7804	7819	7957	8027	8104	8759	8787	8811	8818		
L.CS2	001610	878#	1231*	1271*	1338*	1419*	1429*	1459*	1509*	1552*	1596*	1774*	1784*	1822*
		1833#	1872*	1938*	1980*	2043*	2060*	2104*	2121*	4415*	4443*	4865*	5425*	5440*
		5525#	5596*	5652*	6093*	6532*	6587*	6657*	6669*	6812*	6833*	6853*	6862*	7903
		8021												
L.DA	001606	875#	7814											
L.DCYL	001614	880#	2123*	4419*	4445*	4870*	6854*	6863*	7813	8109	8628	8751		
L.DS	001606	876#	4903*	8112										
L.DT	001607	877#	2062*	2402	4417*	4444*	4867*	8110	8631	8750				
L.MR1	001616	881#	1461*	1475*	1598*	1642*	1706*	1730*	1785*	1832*	5801*	5816*	6010*	6092*



OPER24	045300	7830	10141	10171*														
OPER26	045315	10142	10174*															
OPER30	045333	10143	10177*															
OPER32	045347	10144	10180*															
OPER34	045373	10145	10184*															
OPER36	045417	10146	10188*															
OPER37	045443	4761	4768	10192*														
OPER40	045477	10197*																
OPER41	045477	5066	5072	5083	5089	5201	5207	5218	5224	10198*								
OPI =	020000	265*	8244	8283														
OPTERR =	000020	8285	10054*															
OPR001	045537	1050	10207*															
OPR002	045566	1053	1064	1081	10211*													
OPR003	045574	1060	10212*															
OPR004	045624	1072	10216*															
OPR005	045646	1026	10219*															
OPR006	045717	1230	10226*															
OPR007	046067	1227	1248	10244*														
OPR008	046157	1307	10254*															
OPR009	046242	1313	10263*															
OPR010	046312	3563	10270*															
OPR011	046444	2907	3055	3204	10286*													
OPR012	046607	2908	10304*															
OPR013	046613	3056	10305*															
OPR014	046617	3205	10306*															
OPR015	046623	2910	3057	3206	10307*													
OPR016	046637	1045	10310*															
OPR017	047423	5382	5541	10374*														
OPR018	047540	6641	10388*															
OPR019	047626	9047	10397*															
OPTFLG	001656	907*	1043	1046*	1228*	1249*	1260*	1265	2904	2909*	3052	3058*	3201	3207*				
		3559	3561	3564*	3577	3590	3597	3610	4560	4572	4579	4608	5311	5379				
		5381*	5538	5540*	7498*	7509*	7521*	7548	7645	7800	7846							
		1096	2005	3616	4613	4627	6552	7484*										
OPTTST	032434	240*																
OR =	000200	203*	1371															
PACK =	000103	913*	3559	4560	7498													
PARBKO =	000200	914*	3577	3590	3597	3610	4572	4579	4608	7509								
PARBK1 =	000100	395	943*															
PARM	001716	289*	1598	5801														
PAT =	000020	11045*																
PATCH	056046	8937	9768*															
PAT02	044120	8942	9788*															
PAT03	044160	8947	9808*															
PAT04	044220	8952	9828*															
PAT05	044260	8957	9848*															
PAT06	044320	8962	9871*															
PAT10	044360	8967	9891*															
PAT11	044420	8972	9911*															
PAT12	044460	8977	9931*															
PAT13	044520	8982	9951*															
PAT14	044560	8987	9971*															
PAT15	044620	8992	8995	9991*														
PAT16	044660	61*	956*	1023*	1096*	2000*	2005*	3616*	4613*	4627*	6552*	6975*	6978*	6988*				
PC =	%000007	6993	7026	7073*	7261*	7278*	7338*	7344*	7441*	7527*	7557*	7584*	7642*	7647*				
		7823*	9104*	9123*	9130*	9137*	9151*	9153*	9325*	9339*	9400*	9445*						





	8452*	8461*	8499*	8501*	8504*	8505*	8506*	8508	8515*	8520*	8538*	8544*	8564*
	8569*	8575*	8736	8750*	8754*	8762*	8764*	8767	8769	8770	8776*	8796*	8797*
	8798*	8799*	8800*	8801*	8802*	8803*	8817*	8820*	8830*	8831	8840*	8856*	8858
	8871*	8894*	8906	8924*	8931*	9005*	9011*	9019*	9030*	9067*	9068*	9071*	9072*
	9169	9182*	9183	9187	9211*	9572	9577*	9585*	9587*	9589*	9592*	9595	9598*
	9627	9650*	9660	9683*									
R2 =%000002	54#	1122*	1124*	1145*	1148*	1526*	1793*	1841*	2004*	2335	5368*	5369*	5371*
	5372*	5377	5535	5751*	5806*	5850*	5886*	5930	5933*	6016*	6029*	6100*	6207*
	6353*	6358*	6363*	6425*	6483*	6484*	6485	6602*	7017	7033*	7035*	7036*	7039*
	7040*	7048*	7049	7051*	7056*	7058*	7063*	7068	7070*	7398*	7416*	7702	7713*
	7727	7778*	7779*	7810	7812*	7813*	7814*	7819*	7833	7840	7894*	7903*	7904*
	7905*	7909	7911	7913	7919	7920	7921*	7962*	7964	9066*	9070*	9170	9181*
	9185*	9188	9195*	9196*	9197	9202*	9210*	9573	9578*	9586*	9588*	9590*	9596
	9597*	9628	9649*	9661	9682*								
R3 =%000003	55#	1031*	1039*	1042	1216*	1222*	1286*	1305	1315	1345	2170*	2241*	2242*
	2254*	2255	2260*	2264*	2269*	2270	2274*	2279*	2284	2335*	2336*	2357	2521*
	2522	2524	2526	2529*	5347*	5367*	5391*	5436*	5480*	6521*	6524	6576*	6579
	6648*	6724	6728*	6729	6734	6749	6750*	6759*	6760	6762	6777*	6803*	6877
	6904	6908*	6909	6914	6929	6930*	6939*	6940	6942	6957*	7018	7034*	7038*
	7069*	7400*	7402	7408*	7725	7728*	7730*	7734*	7736*	7738*	7748	7750*	7754*
	7756*	7810*	7811*	7836	7837	7838	7906*	7907*	8020	8027*	8028*	8032*	8036*
	8039*	8045*	8048*	8049	8055*	8056*	8057	8060	8061*	8103*	8106*	8116	8238*
	8276	8279	8297	8305	8309	8313	8317	8321	8353*	8355	8391*	8399*	8408*
	8416*	8428*	8438*	8449*	8454	8456*	8462*	8502	8510	8534	8554	8556	8558
	8580	8589	8593	8596	8737	8751*	8766	8768	8771	8824*	8829*	8839*	8855
	8857*	8859	8870*	8883*	8892*	8896	8897*	8905	8937*	8942*	8947*	8952*	8957*
	8962*	8967*	8972*	8977*	8982*	8987*	8992*	8995*	9004	9018	9029*	9171	9179*
	9180*	9194*	9197*	9206*	9207*	9209*	9255	9264*	9270*	9271*	9274*	9279*	9280*
	9281	9290*	9495	9497*	9498	9501*	9502	9509*	9510	9512	9520	9524	9526*
	9532	9534	9536*	9539*	9629	9648*	9662	9681*					
R4 =%000004	56#	1032*	1036*	1041	1789*	1837*	2175*	2310*	2318*	2338*	2379*	2386*	2394*
	2434*	2443*	2462*	2509*	2517*	2542*	2587*	2595*	2607*	2638*	2646*	2659*	2691*
	2699*	2719*	2751*	2759*	2777*	2811*	2819*	2840*	2848*	2859*	2874*	2883*	2917*
	2924*	2937*	2941*	2953*	2962*	2992*	2999*	3009*	3022*	3031*	3066*	3071*	3085*
	3087*	3100*	3110*	3138*	3145*	3155*	3169*	3179*	3215*	3222*	3235*	3250*	3260*
	3290*	3297*	3307*	3321*	3331*	3352*	3360*	3371*	3380*	3396*	3405*	3427*	3434*
	3445*	3452*	3467*	3476*	3496*	3524*	3568*	3581*	3641*	3649*	3660*	3664*	3679*
	3694*	3703*	3725*	3733*	3748*	3756*	3767*	3782*	3791*	3811*	3819*	3835*	3839*
	3853*	3865*	3874*	3896*	3904*	3916*	3920*	3935*	3950*	3959*	3981*	3989*	4001*
	4005*	4020*	4035*	4044*	4071*	4075*	4079*	4094*	4103*	4129*	4133*	4136*	4151*
	4160*	4186*	4190*	4194*	4209*	4218*	4242*	4246*	4250*	4265*	4274*	4298*	4302*
	4306*	4321*	4330*	4350*	4377*	4421*	4447*	4455*	4463*	4490*	4494*	4498*	4513*
	4528*	4537*	4563*	4589*	4631*	4635*	4639*	4654*	4669*	4678*	4708*	4716*	4729*
	4733*	4737*	4755*	4778*	4793*	4802*	4809*	4813*	4831*	4840*	4872*	4895*	4936*
	4944*	4955*	4993*	4997*	5012*	5032*	5126*	5130*	5145*	5167*	5252*	5256*	5273*
	5317*	5325*	5351*	5394*	5410*	5417*	5445*	5462*	5471*	5518*	5549*	5571*	5586*
	5610*	5630*	5645*	5683*	5704*	5719*	5731*	5775*	5794*	5808*	5822*	5876*	5914*
	5918*	5935*	5976*	5991*	6003*	6013*	6035*	6067*	6085*	6097*	6123*	6139*	6154*
	6178*	6193*	6204*	6239*	6276*	6298*	6313*	6328*	6340*	6350*	6355*	6360*	6371*
	6394*	6432*	6447*	6450*	6635*	6649*	6734*	6736*	6737	6739	6750	6752*	6767*
	6768	6770	6777	6804*	6914*	6916*	6917	6919	6930	6932*	6947*	6948	6950
	6957	7396*	7404	7415*	7420	7683	7688	7691*	7758*	7772	7773	7784*	7809*
	7836*	7837*	7838*	7851*	7852*	7853*	7865	7881*	7933	7934*	7999	8005*	8064*
	8110*	8111*	8119*	8120	8122*	8126	8131	8145*	8210*	8212*	8213*	8239*	8250
	8254	8258	8342*	8346	8357*	8387*	8395*	8405*	8409*	8413*	8425*	8429*	8435*
	8446*	8450*	8457*	8563	8623	8624*	8626	8628	8630*	8631	8634	8638*	8639

R5 =%000005

8644*	8645	8646*	8739	8746	8758*	8761*	8774*	8791*	8822*	8828*	8837	8846*
8852	8854	8865*	8866	8873*	8879	8887	8894	8895	8896*	8904	8917	8918*
8999	9002*	9010	9015*	9028*	9033*	9052*	9256	9258*	9259*	9260*	9261	9262*
9276	9278*	9286*	9289*	9630	9647*	9663	9680*					
57*	2169*	2243*	2244	2262	2263*	2275	6645*	6724	6737	6760	6768	6800*
6904	6917	6940	6948	7808*	7855*	8112*	8113*	8115*	8116	8118*	8127	8134
8240*	8242	8268	8272	8283	8287	8291	8325	8343*	8345	8356*	8358	8388*
8398*	8417*	8439*	8530*	8563*	8738	8752*	8755*	8765*	8766*	8767*	8769*	8771*
8772*	8793*	8804*	8823*	8827*	8837	8851	8853	8869*	8882*	8890*	8903	8918
8919*	8923*	8930*	9002	9004*	9010*	9018*	9027*	9172	9174*	9176*	9183*	9187*
9202	9208*	9257	9263*	9265*	9267*	9268*	9269*	9270	9288*	9631	9646*	9654
9679*												

R6 =%000006

R7 =%000007

SAVREG= 104413

58#	60	959*	960*	961								
59#	61											
7374	7484	7596	7599	7602	7605	7608	7611	7614	7617	7620	7623	7626
7629	7632	7635	7799	7893	8080	8161	8170	9739#				

SCLR = 000040

SCOP1 = 104415

238#	6602	7962										
2078	2086	2188	2332	2364	2371	2400	2483	2563	2935	4431	4454	4470
4773	4882	4912	5027	5050	5094	5160	5185	5229	5366	5389	5435	5460
5478	6703	6883	9741#									

SCOP1\$ 032652

SECERR= 000040

SEEK = 000117

SELDRV= 000101

7535#	9741											
4765	8136	10074#										
209#	1939	1981	2061	2122	5395	5550	5611	5684	5795	5977	6855	
202#	1233	1272	1340	1420	1460	1510	1553	1597	1663	1704	1775	1823
1909	6068	6086	6240	6329	6372	6395	6671	9053				

SETCLK 001706

SKI = 000002

SKIERR= 000002

SP =%000006

397	940#											
253#												
2082	5835	10033#										
60#	951*	952*	953*	963*	981*	989*	993	1051*	1055	1061*	1062*	1066

1073*	1074*	1075*	1076*	1077*	1078*	1079*	1083	1102*	1103*	1115*	1117*	1118*
1135	1136	1209*	1210*	3585*	3586*	3601*	3602*	4567*	4568*	4583*	4584*	6634*
6636	6748*	6749*	6752	6753	6928*	6929*	6932	6933	6982*	7015*	7016*	7017*
7018*	7019*	7020*	7021	7024	7032*	7036	7037*	7044	7064*	7065	7066	7069
7070	7071	7072	7094	7102*	7105	7107	7108	7113*	7114*	7116	7118	7119*
7120*	7121*	7122	7124	7147	7148	7152*	7238*	7239*	7246	7247*	7258	7259*
7260*	7270	7271*	7276	7277	7310	7315	7318	7321*	7324*	7325	7326*	7329
7354*	7357*	7406*	7437*	7468*	7486*	7487*	7524	7525	7539*	7573*	7583	7637*
7638*	7660*	7679*	7680*	7689	7690	7699*	7700*	7714	7715	7723*	7724*	7725*
7738	7739	7740	7747	7748*	7756	7757*	7770*	7771*	7782	7783	7827*	7868*
7873*	7954*	7955*	7979	7980	7986	7987	7988*	8018*	8019*	8020*	8061	8062
8063	8162	8166*	8227*	8231*	8394*	8396	8398	8401	8404*	8406	8409	8411
8412*	8414	8417	8421	8424*	8426	8429	8433	8434*	8436	8439	8442	8445*
8447	8450	8453	8585*	8588*	8591*	8595*	8598*	8622	8633*	8634*	8642*	8643*
8735*	8736*	8737*	8738*	8746*	8793	8796	8799	8800	8846	8865	8869	8870
8871	8872	8879*	8895*	8917*	8919	8999*	9000*	9014	9015	9033	9097*	9098
9108*	9110	9111	9112*	9114	9116	9118	9124	9126*	9128*	9136*	9140	9144
9145	9149	9168*	9169*	9170*	9171*	9172*	9173*	9174	9177*	9190	9192*	9194
9204	9206	9208	9209	9210	9211	9212	9214*	9215*	9247*	9248	9249	9250*
9255*	9256*	9257*	9263	9288	9289	9290	9291*	9292*	9334*	9335*	9336	9340
9342	9351	9353	9356	9359*	9360*	9361	9366	9368	9370*	9371	9387*	9388*
9389	9399	9406*	9409*	9410*	9414*	9415*	9417	9420*	9427	9430*	9434	9436
9438	9439*	9446	9448	9450*	9451	9453*	9454*	9455*	9456*	9457*	9472*	9473*
9474*	9475*	9476*	9482*	9495*	9496*	9501	9504	9508*	9515	9519*	9538	9539
9540*	9541*	9542*	9569*	9570*	9571*	9572*	9573*	9575	9579*	9581	9583	9591*
9592	9594	9595*	9597	9598	9599	9601	9626*	9627*	9628*	9629*	9630*	9631*
9632*	9633*	9634*	9635*	9642*	9643*	9644*	9645*	9646	9647	9648	9649	9650













WCKERR= 000004	5046	5181	8278	10052#										
WLE = 004000	263#													
WLERR = 004000	10043#													
WRDATA= 000123	211#	2463	2543	2608	2660	2720	2778	2841	2918	2993	3072	3139	3216	
	3291	3353	3428	3497	3569	3642	3726	3812	3897	3982	4499	4590	4640	
	4738	4937	4998	5131	5257	5352	5372	5418	5519	5597	5823	5919	6194	
	6866													
WRDCNT 001672	928#	8830	8858*	8906*	9030									
WRDNUM 001670	927#	8829	8859*	8905*	9029									
WRHEAD= 000127	213#	2311	2435	2510	2588	2639	2692	2752	2812	4414	4709	4866	5318	
	6341	6451												
WRL = 004000	280#	1283												
WRTCHK= 000131	214#	4956	5013	5033	5146	5168	5274	5646						
WRTGAT= 040000	299#													
\$APTHD 000224	407	413#												
\$ASTAT= ***** U	7264	7279												
\$ATYC 031306	7235	7237#												
\$ATY1 031262	7233#													
\$ATY3 031270	7234#	9104												
\$ATY4 031300	7236#	7344												
\$AUTOB 001134	444#	1016*	9397	9554										
\$BASE 001352	542#	1051	1059*	1122										
\$BDADR 001122	439#													
\$BDDAT 001126	441#													
\$BELL 001266	493#	7308	7364	9350	9547									
\$CDW1 001356	544#													
\$CHARC 041434	9121*	9131*	9138	9147*	9152#									
\$CKSWR 042410	9383#	9735												
\$CMTAG 001100	427#	958	959	967	973	974	975							
\$CM1 = 000020	459#	460#	461#	462#	463#	464#	465#	466#	467#	468#	469#	470#	471#	
	472#	473#	474#	475#										
\$CM2 = 000040	459#	460#	461#	462#	463#	464#	465#	466#	467#	468#	469#	470#	471#	
	472#	473#	474#	475#										
\$CM3 = 000020	457#	459												
\$CM4 = 000020	475#	476#	477#	478#	479#	480#	481#	482#	483#	484#	485#	486#	487#	
	488#	489#	490#	491#										
\$CNTLC 043306	9338	9419	9547#											
\$CNTLG 043320	9404	9549#												
\$CNTLU 043313	9429	9522	9548#											
\$SCORE 030414	7028	7055#												
\$CPUOP 001324	516#													
\$CRLF 001273	495#	7328	7339	7364	7386	7391	7399	7412	7425	8527	8560	9120	9155	
	9440	9527	9547	9608										
\$CROUT 030444	7055	7062#												
\$DBLK 041654	9179	9213	9221#											
\$DEVCT 001306	507#													
\$DEVN 001354	543#	1216	2199*	2248*	6499	6521	6576	6634	6645	6800				
\$DOAGN 030162	6977	6986	6992#											
\$DTBL 041644	9182	9217#												
\$ENDAD 030152	1004	6988#	7359											
\$ENDCT 030120	973	6979#												
\$ENDMG 030171	6981	6996#												
\$ENULL 030166	6984	6995#												
\$ENV 001316	512#	1010	7242	7266	7341	9099								
\$ENVN 001317	513#	996	1021	1219	7244	9101	9106							
\$EOP 030064	1028	1133	1161	1309	6642	6969#	7228	7439	7469					





		938#	964	965	967	969	971	973	974	975	977	1004	1007	6971
\$SETUP=	000137	7091	7302	7351	7359	9342	9347	9342	9378	9554				
\$SIZE	030206	1023	7015#											
\$SIZEX	030450	7053	7063#											
\$STUP	= 177777	938#												
\$SVLAD	030766	7106	7145#											
\$SWR	= 167400	8#	20	24	25	26	27	28	29	30	31	491	492	493
		974	775	977	978	1114	1144	1205	1415	1455	1505	1548	1592	1770
		1818	1868	1934	1976	2036	2099	2162	2304	2430	2505	2583	2634	2687
		2746	2804	2900	2981	3049	3128	3198	3279	3348	3423	3492	3520	3555
		3637	3721	3807	3892	3977	4064	4123	4180	4237	4293	4346	4373	4408
		4486	4559	4626	4704	4857	4932	4988	5121	5248	5310	5512	5679	5770
		5872	5910	5971	6063	6119	6272	6478	6518	6573	6633	6797	6966	6972
		6987	6993	6995	7082	7083	7084	7085	7086	7092	7109	7111	7112	7125
		7126	7127	7134	7135	7136	7148	7151	7154	7293	7294	7295	7296	7297
		7306	7313	7348	7352	7364	9689							
\$SWREG	001320	514#	998											
\$SWRMK=	000000	31	32	7086	7087	7115								
\$SWOBT	031040	7121	7155#	7657	7877	7977	8602							
\$TERM	= 000102	9761#												
\$TESTN	001302	505#	7146*	7375*	7376*	7655	7875	7975	8600	10931	10934	10940	10948	
\$TIMES	001262	491#	974*	1114*	1144*	1205*	1415*	1455*	1505*	1548*	1592*	1770*	1818*	1868*
		1934*	1976*	2036*	2099*	2162*	2304*	2430*	2505*	2583*	2634*	2687*	2746*	2804*
		2900*	2981*	3049*	3128*	3198*	3279*	3348*	3423*	3492*	3520*	3555*	3637*	3721*
		3807*	3892*	3977*	4064*	4123*	4180*	4237*	4293*	4346*	4373*	4408*	4486*	4559*
		4626*	4704*	4857*	4932*	4988*	5121*	5248*	5310*	5512*	5679*	5770*	5972*	5910*
		5971*	6063*	6119*	6272*	6478*	6493*	6518*	6573*	6633*	6797*	6972*	7134*	7141
		7144*	7154											
\$TKB	001146	450#	9302	9323	9334	9359	9387	9414						
\$TKCNT	042112	9303#	9318*	9348	9365*	9479	9481*							
\$TKINT	042122	956	9318#	9339	9400									
\$TKQEN=	042121	9307#	9373	9484										
\$TKQIN	042114	9304#	9319*	9320	9371*	9372*	9373	9375*						
\$TKQOU	042116	9305#	9320*	9482	9483*	9484	9486*							
\$TKQSR	042120	9306#	9319	9375	9486									
\$TKS	001144	449#	9302	9324*	9355*	9357	9363*	9385	9401*	9411	9423*	9443*		
\$TKSRV	042172	9321	9334#											
\$TMP0	001222	475#	1055*	1057	1059	1066*	1068	1071	1083*	1085	1087	1089*	1090*	1091*
		1092*	1093*	1095	7963*	7967*								
\$TMP1	001224	476#	6644*	6720*	6745	6756	6778*	6799*	6900*	6925	6936	6958*	8389	8768*
		8770*	8772	8786*	8802	8803	8807	8809*						
\$TMP10	001242	483#	8163*	8343	8445	8575								
\$TMP11	001244	484#	8164*	8424	8569									
\$TMP12	001246	485#	8165*	8404	8564									
\$TMP13	001250	486#	8167*	8171*	8201	8340	8358	8402	8422	8443	8494	8549		
\$TMP14	001252	487#	8517*	8536	8540*	8542	8546*	8567	8571*	8573	9577*			
\$TMP15	001254	488#	8518*	8523*										
\$TMP16	001256	489#	7747*	7752*										
\$TMP17	001260	490#	7749*	7757	7759*									
\$TMP2	001226	477#	8390											
\$TMP3	001230	478#	7902*	7916*	7931									
\$TMP4	001232	479#	1232*	1251	1255*									
\$TMP5	001234	480#	8625*	8635*	8642									
\$TMP6	001236	481#	8622*	8643										
\$TMP7	001240	482#	1332*	1396*	1399	8623*	8644							
\$TN	= 000111	9#	20	1108	1114#	1138	1144#	1167	1205#	1407	1415#	1444	1455#	1496





COMMEN	1#	142#															
ENDCOM	1#	142#															
ERROR	36#	1131	1159	1207	1237	1246	1269	1276	1298	1336	1344	1352	1357	1364	1369		
	1375	1378	1383	1390	1417	1424	1427	1433	1436	1440	1457	1464	1466	1473	1479		
	1482	1489	1507	1514	1517	1524	1535	1550	1558	1564	1571	1594	1602	1608	1622		
	1627	1633	1640	1649	1655	1661	1667	1673	1678	1684	1690	1696	1709	1712	1718		
	1724	1736	1739	1750	1756	1772	1779	1782	1796	1802	1805	1820	1827	1830	1844		
	1850	1870	1877	1884	1892	1899	1906	1913	1917	1923	1936	1943	1952	1964	1967		
	1978	1985	1993	2012	2019	2040	2048	2053	2058	2066	2077	2085	2101	2109	2114		
	2119	2127	2133	2173	2185	2187	2198	2208	2218	2225	2234	2247	2289	2308	2324		
	2327	2330	2349	2352	2355	2363	2370	2378	2381	2382	2383	2389	2398	2432	2455		
	2458	2461	2472	2481	2491	2507	2534	2536	2540	2552	2561	2570	2585	2602	2605		
	2617	2623	2636	2654	2657	2669	2675	2689	2714	2717	2729	2735	2749	2772	2775		
	2786	2789	2806	2824	2827	2854	2857	2869	2872	2878	2886	2915	2931	2934	2950		
	2952	2958	2965	2991	3005	3008	3019	3021	3027	3034	3065	3081	3084	3096	3099		
	3105	3113	3137	3151	3154	3165	3168	3174	3182	3214	3229	3232	3233	3245	3248		
	3255	3263	3288	3303	3306	3317	3320	3326	3334	3350	3366	3369	3386	3389	3400		
	3408	3425	3440	3443	3458	3460	3471	3479	3494	3506	3511	3522	3534	3539	3557		
	3608	3622	3639	3655	3658	3674	3677	3689	3692	3698	3706	3723	3739	3742	3762		
	3765	3777	3780	3786	3794	3809	3825	3828	3849	3852	3863	3869	3877	3894	3910		
	3913	3930	3933	3945	3948	3954	3962	3979	3995	3999	4015	4018	4030	4033	4039		
	4047	4066	4089	4092	4098	4106	4125	4146	4149	4155	4163	4182	4204	4207	4213		
	4221	4239	4260	4263	4269	4277	4295	4316	4319	4325	4333	4348	4360	4363	4375		
	4387	4393	4413	4426	4429	4441	4449	4450	4451	4458	4466	4488	4508	4511	4523		
	4526	4532	4540	4588	4599	4602	4629	4649	4652	4664	4667	4673	4681	4706	4724		
	4727	4747	4753	4762	4771	4777	4788	4791	4797	4805	4823	4829	4835	4843	4863		
	4877	4880	4893	4907	4910	4934	4950	4953	4965	4968	4992	5007	5010	5022	5025		
	5031	5042	5048	5067	5075	5084	5092	5125	5140	5143	5155	5158	5165	5177	5183		
	5202	5210	5219	5227	5250	5266	5269	5283	5286	5330	5333	5350	5361	5364	5375		
	5383	5388	5404	5407	5409	5416	5429	5433	5444	5455	5458	5466	5474	5514	5533		
	5542	5547	5559	5563	5566	5569	5581	5584	5605	5609	5619	5623	5626	5628	5640		
	5643	5662	5666	5681	5693	5696	5700	5702	5714	5717	5729	5741	5747	5749	5753		
	5773	5785	5789	5792	5804	5820	5832	5838	5848	5859	5874	5889	5895	5912	5945		
	5951	5956	5974	5986	5989	6001	6019	6024	6028	6033	6045	6047	6065	6077	6080		
	6083	6103	6106	6121	6133	6137	6149	6152	6165	6168	6173	6176	6188	6191	6209		
	6233	6237	6250	6256	6274	6286	6289	6293	6296	6308	6311	6323	6326	6338	6366		
	6369	6383	6389	6392	6407	6410	6416	6423	6428	6442	6445	6460	6463	6481	6489		
	6523	6537	6541	6544	6561	6578	6591	6595	6598	6614	6655	6663	6665	6675	6678		
	6684	6693	6701	6809	6817	6821	6826	6829	6838	6841	6845	6848	6859	6871	6874		
	6881	9050	9061	9064													
ESCAPE	1#	142#															
GETPRI	1#	142#	7023														
GETSWR	1#	142#	1007#														
INITSS	934#	2322	2347	2376	2913	2989	3063	3135	3212	3286	3424	3493	3521	3556	3638		
	3722	3808	3893	3978	4065	4124	4181	4238	4294	4347	4374	4411	4439	4487	4587		
	4628	4705	4775	4861	4891	4933	4991	5030	5124	5164	5249	5348	5414	5442	5513		
	5680	5748	5772	5847	5873	5911	5973	6026	6064	6082	6120	6273	6391	6415	6422		
	6480	6488	6522	6577	6653	6807	9049										
LOADLS	936#	2941	2992	3087	3138	3235	3290	3427	3445	3567	3641	3664	3679	3725	3748		
	3767	3811	3839	3853	3896	3920	3935	3981	4005	4020	4079	4136	4194	4250	4306		
	4350	4377	4498	4513	4589	4639	4654	4708	4737	4778	4813	4894	4936	4955	4997		
	5012	5032	5130	5145	5167	5256	5273	5351	5394	5417	5518	5549	5571	5586	5610		
	5630	5645	5683	5704	5719	5731	5775	5794	5808	5822	5876	5918	5935	5976	5991		
	6003	6035	6067	6085	6123	6139	6154	6178	6193	6239	6276	6298	6313	6328	6340		
	6371	6394	6432	6450	9052												
MSG	1108#	1110	1138#	1140	1166#	1169	1407#	1409	1444#	1446	1496#	1498	1540#	1542	1573#		

	1575	1759#	1761	1806#	1808	1854#	1856	1925#	1927	1968#	1970	2025#	2027	2091#	2093
	2137#	2139	2294#	2296	2413#	2415	2494#	2496	2573#	2575	2625#	2627	2676#	2678	2736#
	2738	2794#	2796	2889#	2891	2970#	2972	3038#	3040	3117#	3119	3187#	3189	3268#	3270
	3339#	3341	3412#	3414	3484#	3486	3512#	3514	3540#	3542	3628#	3630	3711#	3713	3798#
	3800	3882#	3884	3967#	3969	4052#	4054	4111#	4113	4168#	4170	4225#	4227	4281#	4283
	4337#	4339	4364#	4366	4398#	4400	4477#	4479	4545#	4547	4616#	4618	4689#	4691	4848#
	4850	4923#	4925	4969#	4971	5102#	5104	5237#	5239	5290#	5292	5495#	5497	5669#	5671
	5759#	5762	5863#	5865	5897#	5900	5959#	5962	6051#	6053	6107#	6109	6257#	6259	6465#
	6467	6510#	6512	6564#	6566	6617#	6619	6781#	6783						
MULT	1#	142#													
NEWTST	1#	142#	1108	1138	1167	1407	1444	1496	1540	1573	1759	1806	1854	1925	1968
	2025	2091	2137	2294	2413	2494	2573	2625	2676	2736	2794	2889	2970	3038	3117
	3187	3268	3339	3412	3484	3512	3540	3628	3711	3798	3882	3967	4052	4111	4168
	4225	4281	4337	4364	4398	4477	4545	4616	4689	4848	4923	4969	5102	5237	5290
OPCHK	5495	5669	5760	5863	5898	5960	6051	6107	6257	6465	6510	6564	6617	6781	6781
	938#	2933	2951	3007	3020	3083	3098	3153	3167	3231	3247	3305	3319	3442	3459
	3657	3676	3691	3741	3764	3779	3827	3851	3912	3932	3947	3998	4017	4032	4091
	4148	4206	4262	4318	4362	4428	4510	4525	4601	4651	4666	4726	4790	4879	4909
	4952	4967	5009	5024	5142	5157	5268	5285	5332	5363	5386	5406	5431	5457	5545
	5562	5568	5583	5607	5622	5627	5642	5664	5695	5701	5716	5791	5954	6046	6079
	6105	6151	6167	6175	6190	6232	6288	6295	6310	6325	6368	6409	6444	6462	6543
POP	6597	6664	6677	6827	6846	6873	9063								
	1#	142#	7276	7277	7689	7782	7979	8845	8864	8868	9032	9208	9597	9646	9678
PUSH	9679														
REPORT	1#	142#	7237	7239	7260	7678	7769	7953	8734	8745	9167	9571	9626	9659	9665
RESDC	1#	142#													
	935#	2959	3028	3107	3176	3257	3328	3473	3700	3788	3871	3956	4041	4100	4157
RKLOAD	4215	4271	4327	4460	4534	4675	4799	4837	5468						
	937#	2929	2948	3003	3017	3079	3094	3149	3163	3227	3243	3301	3315	3438	3456
	3504	3532	3606	3653	3672	3687	3737	3760	3775	3823	3847	3861	3908	3928	3943
	3993	4013	4028	4087	4144	4202	4258	4314	4358	4385	4506	4521	4597	4647	4662
	4722	4745	4786	4821	4875	4905	4948	4963	5005	5020	5040	5138	5153	5175	5264
	5281	5328	5359	5402	5453	5557	5579	5617	5638	5691	5712	5727	5739	5783	5802
	5818	5830	5943	5984	5999	6043	6075	6131	6147	6163	6186	6248	6284	6306	6321
	6336	6381	6404	6440	6458	6535	6589	6661	6673	6815	6824	6836	6843	6857	
SCOPE	37#	1113	1143	1204	1414	1454	1504	1547	1591	1769	1817	1867	1933	1975	2035
	2098	2161	2303	2429	2504	2582	2633	2686	2745	2803	2899	2980	3048	3127	3197
	3278	3347	3422	3491	3519	3554	3636	3720	3806	3891	3976	4063	4122	4179	4236
	4292	4345	4372	4407	4485	4558	4625	4703	4856	4931	4987	5120	5247	5309	5511
	5678	5769	5871	5909	5970	6062	6118	6271	6477	6492	6517	6572	6632	6796	6970
SETPRI	1#	142#	9475												
SETTRA	9719#	9728	9729	9730	9731	9733	9735	9736	9737	9738	9739	9740	9741	9742	9743
	9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754	9755	9756	9757	9758
	9759	9760													
SETUP	1#	142#	957												
SKIP	1#	142#													
SLASH	1#	142#													
SPACE	142#														
STARS	1#	142#	400	402	409	422	497	500	1108	1112	1138	1142	1167	1203	1407
	1413	1444	1453	1496	1503	1540	1546	1573	1590	1759	1768	1806	1816	1854	1866
	1925	1932	1968	1974	2025	2034	2091	2097	2137	2160	2294	2302	2413	2428	2494
	2503	2573	2581	2625	2632	2676	2685	2736	2744	2794	2802	2889	2898	2970	2979
	3038	3047	3117	3126	3187	3196	3268	3277	3339	3346	3412	3421	3484	3490	3512
	3518	3540	3553	3628	3635	3711	3719	3798	3805	3882	3890	3967	3975	4052	4062
	4111	4121	4168	4178	4225	4235	4281	4291	4337	4344	4364	4371	4398	4406	4477
	4484	4545	4557	4616	4624	4689	4702	4848	4855	4923	4930	4969	4986	5102	5119



.\$SB2D	1#		
.\$SB20	1#		
.\$SCOP	1#	5#	7076
.\$SIZE	1#	6#	6999
.\$SUPR	1#		
.\$STRAP	1#	6#	9696
.\$STYPB	1#		
.\$STYPD	1#	7#	9155
.\$STYPE	1#	5#	9076
.\$STYPO	1#	5#	9222
.\$4DCA	1#	4#	366
.\$1170	1#		



# K05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(732) 01-OCT-76 10:50 PAGE 239  
 DZR6KB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SEQ 0269

BLE	2567	7417																	
BLO	1086	5931	7649	9511															
BLOS	1069	9499																	
BLT	7117	8207	9129	9184	9200	9285	9367	9447	9584										
BMI	7824	7841	8535	8555	8627	8782	9191												
BNE	962	986	1003	1009	1013	1022	1038	1044	1049	1153	1157	1218	1224	1243	1252				
	1254	1280	1284	1295	1306	1316	1320	1346	1367	1381	1402	1469	1520	1567	1616				
	1619	1624	1630	1636	1652	1658	1675	1681	1687	1693	1715	1721	1728	1743	1881				
	1903	1920	1949	1958	1990	2002	2071	2090	2193	2196	2203	2206	2212	2215	2223				
	2245	2253	2259	2268	2285	2373	2403	2530	2905	3053	3202	3395	3466	3560	3562				
	3834	4434	4472	4885	5097	5232	5312	5380	5392	5485	5489	5532	5539	5604	5661				
	5845	5929	6495	6500	6502	6525	6529	6551	6580	6584	6605	6639	6681	6697	6721				
	6725	6738	6761	6769	6831	6850	6878	6901	6905	6918	6941	6949	7061	7098	7137				
	7243	7249	7252	7269	7314	7342	7360	7405	7464	7497	7508	7519	7549	7553	7576				
	7578	7580	7644	7646	7685	7731	7737	7755	7781	7822	7844	7856	7866	7899	7908				
	7912	7926	7932	7961	7968	7983	8031	8117	8121	8125	8180	8182	8184	8186	8263				
	8348	8359	8495	8503	8524	8531	8537	8568	8583	8629	8632	8742	8748	8756	8780				
	8805	8838	8841	8922	8925	8929	8932	8936	8941	8946	8951	8956	8961	8966	8971				
	8976	8981	8986	8991	8998	9008	9012	9020	9069	9073	9100	9107	9109	9117	9125				
	9139	9146	9189	9275	9337	9343	9349	9354	9362	9374	9384	9390	9418	9422	9428				
	9435	9442	9485	9503	9505	9521	9525	9535	9677										
BPL	1034	1302	1394	7028	7349	8035	8044	9026	9094	9143	9175	9205	9273	9358	9386				
	9412																		
BR	384	941	944	947	988	1015	1018	1229	1250	1256	1292	1321	1328	1362	1384				
	1391	1397	1494	1885	1888	1889	1900	1907	1953	1994	2008	2013	2079	2209	2219				
	2226	2229	2235	2286	2321	2346	2375	2388	2397	2399	2488	2525	2568	2709	2767				
	2834	2838	2877	2885	2887	2911	2957	2964	2966	2985	3026	3033	3035	3059	3104				
	3112	3114	3131	3173	3181	3183	3208	3254	3262	3264	3282	3325	3333	3335	3399				
	3407	3409	3470	3478	3480	3565	3697	3705	3707	3785	3793	3795	3868	3876	3878				
	3953	3961	3963	4038	4046	4048	4097	4105	4107	4154	4162	4164	4212	4220	4222				
	4268	4276	4278	4324	4332	4334	4436	4457	4465	4467	4474	4531	4539	4541	4672				
	4680	4682	4763	4772	4796	4804	4806	4834	4842	4844	4887	4917	5076	5099	5211				
	5234	5338	5344	5465	5473	5475	5492	5528	5529	5600	5601	5657	5658	5854	5855				
	5953	6503	6530	6546	6585	6600	6651	6688	6689	6731	6741	6764	6772	6806	6911				
	6921	6944	6952	7043	7053	7100	7106	7109	7123	7132	7135	7235	7257	7330	7347				
	7411	7426	7551	7598	7601	7604	7607	7610	7613	7616	7619	7622	7625	7628	7631				
	7634	7659	7760	7831	7859	7869	7918	7929	7972	7981	7985	8024	8033	8096	8168				
	8222	8351	8458	8459	8500	8525	8532	8541	8547	8566	8572	8578	8636	8640	8777				
	8783	8814	8847	8880	8891	8926	8933	8938	8943	8948	8953	8958	8963	8968	8973				
	8978	8983	8988	8993	9016	9074	9096	9122	9132	9141	9148	9186	9203	9251	9266				
	9287	9352	9431	9458	9460	9514	9523	9529	9531	9593	9606	9669	9691						
CLC	7751	8519																	
CLR	949	960	974	975	995	1032	1073	1123	1145	1147	1222	1231	1232	1262	1330				
	1461	1642	1730	1833	1878	1996	1999	2050	2111	2168	2169	2263	3393	3576	3579				
	3609	3612	3613	3614	3832	4410	4438	4607	4610	4611	4612	4754	4858	4888	5271				
	5315	5370	5405	5437	5438	5516	5560	5620	5698	5750	5787	5816	5852	5987	6016				
	6030	6093	6134	6170	6207	6231	6246	6291	6363	6379	6418	6430	6507	6508	6519				
	6539	6574	6593	6646	6647	6667	6686	6778	6801	6802	6818	6839	6958	6971	6972				
	7032	7040	7052	7056	7113	7134	7149	7396	7490	7502	7510	7523	7547	7556	7698				
	7749	7802	7817	7818	7858	7902	7927	7959	8087	8118	8122	8171	8174	8175	8176				
	8177	8178	8218	8349	8350	8387	8388	8517	8625	8633	8824	8897	8923	9178	9181				
	9264	9318	9355	9401	9409	9410	9474	9475	9496	9519	9577	9578	9675						
CLRB	2163	2228	2305	2306	2809	2836	5314	5340	5486	5490	7133	7273	7274	7275	9121				
	9147	9207	9526	9536	9602														
CMP	961	985	1004	1012	1024	1048	1057	1068	1085	1087	1225	1401	1491	1727	1960				
	2001	2089	2202	2211	2239	2244	2252	2258	2267	2281	2284	2474	2554	2566	2829				

	4914	5060	5069	5078	5086	5195	5204	5213	5221	5341	5930	6636	7044	7049	7060
	7094	7107	7116	7141	7315	7329	7359	7431	7648	7865	7917	7925	8001	8051	8057
	8092	8116	8120	8128	8206	8208	8347	8358	8542	8573	8628	8837	9199	9336	9342
	9344	9348	9353	9361	9366	9368	9373	9383	9389	9417	9427	9434	9446	9448	9484
	9498	9510													
CMPB	1010	1223	2195	2205	2214	2222	2485	2831	2901	2982	3050	3129	3199	3280	5335
	5484	5488	7127	7242	7341	8131	8134	8631	8921	8928	8935	8940	8945	8950	8955
	8960	8965	8970	8975	8980	8985	8990	9099	9114	9116	9124	9145	9149	9397	9421
	9441	9502	9520	9524	9534	9581	9583								
COM	7415														
DEC	1037	1318	1615	1742	1957	2372	2392	2881	2960	3029	3108	3177	3258	3329	3394
	3403	3465	3474	3701	3789	3833	3872	3957	4042	4101	4158	4216	4272	4328	4461
	4535	4676	4800	4838	5367	5391	5469	5480	5531	5603	5660	5844	6550	6604	6975
	7119	7379	7408	7416	7577	7579	7684	7730	7736	7754	7780	7843	7855	7898	7907
	7960	7967	8045	8114	8124	8523	8530	8638	8755	8774	8804	8840	8844	8856	8862
	8924	8931	9005	9007	9011	9019	9024	9068	9072	9481	9509				
DECB	9128	9131	9272	9283											
EMT	36														
HALT	383	7350	7361	7440	9095	9668	9690								
INC	1002	1288	1303	1358	1493	1726	2088	2243	2251	2837	4435	4473	4886	4916	5343
	5491	6497	6527	6582	6720	6727	6735	6758	6766	6900	6907	6915	6938	6946	6973
	7045	7140	7272	7309	7428	7447	7456	7475	7581	7759	7924	8036	8094	8099	8115
	8119	8123	8540	8546	8571	8577	8630	8635	8776	8836	8839	8857	8914	9185	9278
	9286	9365	9372	9456	9483	9676									
INCB	2487	2565	2833	5337	5483	5487	7145	7303	9151						
IOT	37	388													
JMP	392	393	395	397	1028	1133	1161	1221	1309	1403	2166	2200	2249	2257	2265
	2272	2290	2407	5384	5543	6506	6642	6743	6754	6774	6779	6923	6934	6954	6959
	6993	7096	7439	7469	8190	8203	8493	8744	8749	9031	9034	9341	9424		
JSR	956	1023	1096	1789	1837	2000	2005	2175	2310	2318	2338	2379	2386	2394	2434
	2443	2462	2509	2517	2542	2587	2595	2607	2638	2646	2659	2691	2699	2719	2751
	2759	2777	2811	2819	2840	2848	2859	2874	2883	2917	2924	2937	2941	2953	2962
	2992	2999	3009	3022	3031	3066	3071	3085	3087	3100	3110	3138	3145	3155	3169
	3179	3215	3222	3235	3250	3260	3290	3297	3307	3321	3331	3352	3360	3371	3380
	3396	3405	3427	3434	3445	3452	3467	3476	3496	3524	3568	3581	3616	3641	3649
	3660	3664	3679	3694	3703	3725	3733	3748	3756	3767	3782	3791	3811	3819	3835
	3839	3853	3865	3874	3896	3904	3916	3920	3935	3950	3959	3981	3989	4001	4005
	4020	4035	4044	4071	4075	4079	4094	4103	4129	4133	4136	4151	4160	4186	4190
	4194	4209	4218	4242	4246	4250	4265	4274	4298	4302	4306	4321	4330	4350	4377
	4421	4447	4455	4463	4490	4494	4498	4513	4528	4537	4563	4589	4613	4627	4631
	4635	4639	4654	4669	4678	4708	4716	4729	4733	4737	4755	4778	4793	4802	4809
	4813	4831	4840	4872	4895	4936	4944	4955	4993	4997	5012	5032	5126	5130	5145
	5167	5252	5256	5273	5317	5325	5351	5394	5410	5417	5445	5462	5471	5518	5549
	5571	5586	5610	5630	5645	5683	5704	5719	5731	5775	5794	5808	5822	5876	5914
	5918	5935	5976	5991	6003	6013	6035	6067	6085	6097	6123	6139	6154	6178	6193
	6204	6239	6276	6298	6313	6328	6340	6350	6355	6360	6371	6394	6432	6447	6450
	6552	6635	6988	7261	7338	7344	7642	7647	7823	8210	8212	8213	8395	8405	8413
	8425	8435	8446	8457	8791	9052	9104	9123	9130	9137	9339	9400	9445		
MOV	940	943	946	951	952	953	959	963	965	966	967	968	969	970	971
	972	973	977	978	981	982	983	984	989	991	992	993	998	1027	1029
	1030	1031	1042	1051	1055	1059	1061	1066	1083	1097	1099	1100	1102	1103	1114
	1115	1116	1117	1118	1119	1120	1121	1122	1124	1130	1132	1134	1135	1136	1144
	1145	1148	1158	1160	1205	1209	1210	1213	1214	1215	1216	1233	1255	1263	1271
	1272	1308	1311	1312	1323	1331	1332	1333	1338	1339	1340	1356	1368	1371	1382
	1388	1389	1395	1396	1399	1415	1419	1420	1429	1439	1455	1458	1459	1460	1471
	1472	1475	1487	1488	1505	1509	1510	1522	1523	1526	1533	1534	1548	1552	1553



# MOS

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(732) 01-OCT-76 10:50 PAGE 241  
DZR6KB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SEQ 0271

1569	1570	1592	1596	1597	1598	1610	1612	1620	1621	1625	1626	1631	1632	1637
1638	1653	1654	1659	1660	1663	1676	1677	1682	1683	1688	1689	1694	1695	1703
1704	1706	1716	1717	1722	1723	1732	1741	1748	1749	1754	1755	1770	1774	1775
1785	1818	1822	1823	1832	1868	1872	1873	1882	1883	1891	1897	1898	1904	1905
1909	1921	1922	1934	1938	1939	1950	1951	1956	1962	1963	1976	1980	1981	1991
1992	2004	2010	2011	2017	2018	2036	2037	2042	2043	2055	2060	2061	2099	2103
2104	2116	2121	2122	2123	2162	2170	2171	2197	2207	2217	2224	2233	2237	2241
2242	2246	2254	2255	2260	2262	2264	2269	2270	2274	2275	2278	2279	2288	2304
2320	2334	2335	2337	2356	2357	2361	2362	2368	2369	2374	2391	2430	2446	2459
2505	2520	2521	2538	2583	2634	2687	2702	2746	2762	2763	2804	2808	2851	2880
2900	2912	2959	2981	2987	3028	3049	3061	3107	3128	3133	3176	3198	3210	3257
3279	3284	3328	3348	3391	3392	3402	3423	3462	3463	3464	3473	3492	3520	3555
3585	3586	3592	3593	3594	3595	3596	3599	3601	3602	3637	3700	3721	3788	3807
3830	3831	3871	3892	3956	3977	4041	4064	4100	4123	4157	4180	4215	4237	4271
4293	4327	4346	4373	4408	4409	4414	4415	4416	4418	4419	4437	4442	4443	4445
4460	4486	4534	4559	4567	4568	4574	4575	4576	4577	4578	4581	4583	4584	4626
4675	4704	4758	4759	4760	4761	4767	4768	4769	4770	4774	4799	4837	4857	4859
4865	4866	4868	4869	4870	4889	4932	4988	4989	5028	5029	5063	5064	5065	5066
5071	5072	5073	5074	5080	5081	5082	5083	5088	5089	5090	5091	5121	5122	5161
5163	5198	5199	5200	5201	5206	5207	5208	5209	5215	5216	5217	5218	5223	5224
5225	5226	5248	5310	5346	5347	5368	5369	5371	5372	5382	5436	5439	5468	5512
5515	5541	5594	5595	5653	5679	5751	5770	5801	5806	5843	5850	5857	5872	5886
5910	5971	6010	6029	6063	6092	6119	6200	6272	6347	6402	6425	6478	6483	6484
6493	6518	6520	6521	6532	6533	6549	6559	6560	6573	6575	6576	6587	6588	6603
6612	6613	6633	6634	6644	6645	6648	6649	6650	6657	6658	6659	6669	6670	6671
6682	6683	6691	6692	6699	6700	6733	6734	6748	6749	6750	6751	6752	6753	6776
6777	6797	6799	6800	6803	6804	6805	6811	6812	6813	6823	6832	6833	6834	6842
6852	6853	6854	6855	6861	6862	6863	6864	6865	6866	6876	6879	6880	6913	6914
6928	6929	6930	6931	6932	6933	6956	6957	6978	6982	6985	7015	7016	7017	7018
7019	7020	7021	7024	7025	7029	7033	7034	7035	7036	7039	7041	7042	7046	7051
7055	7064	7065	7066	7067	7068	7069	7070	7071	7072	7102	7103	7105	7108	7118
7122	7131	7143	7144	7147	7148	7151	7152	7238	7239	7246	7250	7255	7256	7258
7260	7270	7276	7277	7305	7310	7318	7321	7325	7326	7354	7357	7384	7389	7394
7397	7398	7406	7418	7437	7438	7467	7468	7485	7486	7487	7488	7489	7491	7499
7500	7501	7503	7511	7512	7513	7514	7522	7524	7525	7539	7550	7573	7574	7582
7583	7597	7600	7603	7606	7609	7612	7615	7618	7621	7624	7627	7630	7633	7636
7637	7638	7641	7651	7653	7654	7655	7657	7679	7680	7681	7682	7683	7686	7688
7689	7690	7699	7700	7701	7702	7704	7705	7706	7707	7709	7710	7712	7713	7714
7715	7723	7724	7725	7726	7727	7728	7729	7734	7735	7738	7739	7740	7747	7748
7750	7756	7757	7770	7771	7782	7783	7803	7806	7807	7808	7809	7810	7812	7813
7814	7816	7819	7829	7830	7836	7837	7838	7848	7851	7852	7853	7857	7861	7862
7863	7864	7871	7872	7875	7877	7894	7895	7896	7897	7900	7901	7903	7904	7905
7906	7916	7919	7920	7921	7928	7954	7955	7956	7957	7958	7962	7963	7964	7971
7973	7975	7977	7979	7980	7984	7986	7987	7999	8004	8018	8019	8020	8023	8025
8040	8049	8050	8054	8060	8061	8062	8063	8081	8083	8103	8106	8109	8126	8127
8137	8138	8139	8140	8162	8163	8164	8165	8166	8167	8204	8216	8217	8223	8224
8225	8227	8236	8237	8238	8239	8240	8266	8295	8342	8343	8345	8346	8353	8356
8357	8389	8390	8391	8392	8393	8394	8404	8412	8418	8419	8424	8430	8431	8434
8440	8441	8445	8451	8452	8460	8461	8499	8501	8504	8505	8508	8515	8516	8518
8526	8538	8539	8544	8545	8563	8564	8565	8569	8570	8575	8576	8600	8602	8622
8623	8624	8642	8643	8644	8735	8736	8737	8738	8739	8740	8743	8746	8751	8752
8758	8761	8765	8766	8767	8768	8769	8785	8786	8789	8793	8796	8809	8810	8813
8817	8820	8822	8823	8827	8828	8829	8830	8846	8851	8852	8853	8854	8855	8858
8859	8865	8869	8870	8871	8872	8879	8882	8883	8887	8890	8892	8894	8895	8896
8903	8904	8905	8906	8917	8918	8919	8930	8937	8942	8947	8952	8957	8962	8967
8972	8977	8982	8987	8992	8995	8999	9000	9001	9002	9004	9010	9014	9015	9018

MOV8	9027	9028	9029	9030	9033	9066	9067	9070	9071	9097	9098	9103	9111	9126	9168
	9169	9170	9171	9172	9173	9174	9179	9182	9202	9208	9209	9210	9211	9212	9214
	9215	9247	9255	9256	9257	9263	9270	9288	9289	9290	9291	9292	9319	9320	9321
	9322	9324	9363	9375	9406	9423	9438	9443	9472	9473	9476	9486	9495	9497	9508
	9539	9540	9541	9542	9569	9570	9571	9572	9573	9575	9576	9595	9596	9597	9598
	9599	9626	9627	9628	9629	9630	9631	9632	9633	9634	9635	9642	9643	9644	9645
	9646	9647	9648	9649	9650	9651	9657	9658	9659	9660	9661	9662	9663	9664	9665
	9666	9667	9673	9674	9678	9679	9680	9681	9682	9683	9684	9685	9686	9704	9705
	9709	9715	9716												
	976	1016	1040	1041	1074	1101	2062	2256	2261	2271	2405	2406	4417	4444	4867
	4903	6161	6505	7114	7146	7150	7233	7234	7236	7312	7343	7375	7377	7400	7772
	7773	8027	8055	8110	8112	8141	8143	8588	8591	8595	8598	8634	8750	9108	9136
	9144	9177	9180	9194	9197	9206	9248	9249	9252	9253	9254	9258	9261	9262	9281
NEG	9334	9359	9371	9387	9402	9414	9482	9501	9506	9512	9517	9532	9579	9707	
NOP	8026	8042	8082	8085	9176	9259									
	1125	1126	1127	1149	1150	1151	6420	6989	6990	6991	7492	7493	7494	7504	7505
	7506	7515	7516	7517											
RESET	950	1997	6547	6987											
ROL	9265	9267	9268	9269	9271	9586	9588	9590							
ROR	7752	8520													
RTI	954	990	1104	1211	3587	3603	4569	4585	7153	7363	7448	7457	7477	7540	7639
	7662	7716	7741	7989	8228	8232	8586	8605	9113	9216	9293	9364	9376	9444	9477
	9487	9543	9600	9636	9652	9689	9717								
RTS	7073	7278	7441	7527	7557	7584	7691	7758	7784	7881	7934	8005	8064	8145	8646
	8873	9153	9325	9710											
SBC	8048														
SOB	7038														
SUB	7062	7063	7253	7311	7319	7324	7658	7878	7978	8047	8086	8095	8603	9183	
TRAP	7023	9719	9728	9729	9730	9731	9733	9735	9736	9737	9738	9739	9740	9741	9742
	9743	9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754	9755	9756	9757
	9758	9759	9760												
TST	1008	1033	1128	1152	1217	1251	1301	1305	1393	2164	2192	2231	2449	4433	4471
	4884	5928	6638	7030	7047	7059	7104	7124	7138	7248	7266	7268	7348	7355	7402
	7404	7413	7420	7435	7495	7507	7518	7552	7554	7575	7643	7687	7733	7821	7824
	7833	7840	7922	7931	7933	8030	8034	8043	8097	8142	8185	8201	8214	8221	8340
	8396	8401	8402	8406	8411	8414	8421	8422	8426	8433	8436	8442	8443	8447	8453
	8494	8534	8536	8549	8554	8567	8626	8639	8645	8781	8831	8866	9025	9110	9118
	9140	9188	9198	9276	9323	9340	9351	9356	9399	9436	9451	9479	9504	9515	9538
	9594	9601	9706												
TSTB	1154	2190	2402	6556	6609	7026	7125	7240	7251	7264	7401	7537	9093	9142	9190
	9204	9357	9385	9411											
.ASCII	494	495	10226	10270	10286	10310	10315	10324	10333	10339	10346	10351	10359	10374	10397
	10670	10779													
.ASCIZ	493	496	1020	6996	7331	7332	9547	9548	9549	9550	9552	9693	10149	10152	10154
	10156	10158	10161	10164	10166	10167	10169	10171	10174	10177	10180	10184	10188	10192	10198
	10206	10207	10211	10212	10216	10219	10234	10244	10254	10263	10278	10298	10304	10305	10306
	10307	10367	10381	10388	10407	10414	10426	10435	10444	10451	10461	10468	10476	10482	10486
	10490	10498	10503	10507	10512	10516	10520	10523	10527	10530	10534	10538	10541	10543	10545
	10548	10550	10554	10556	10559	10567	10570	10576	10583	10586	10589	10591	10595	10598	10602
	10607	10610	10614	10620	10623	10630	10635	10644	10652	10659	10675	10681	10689	10698	10702
	10707	10715	10718	10724	10731	10735	10739	10742	10748	10751	10755	10759	10762	10765	10769
	10772	10775	10789	10796	10800	10804	10813	10816	10826	10834	10845	10854	10861	10870	10876
	10878	10885	10889	10894	10898	10902	10907	10918	10922						
.BLKB	9306	9546													
.BLKW	9221	11043	11044	11045	11046	11047									
.BYTE	429	430	435	436	444	445	453	454	455	456	512	513	523	524	531

5332	5334	5335	5337	5338	876	877	1790	1791	1838	1839	2179	2180	2314	2315	
2342	2343	2438	2439	2466	2467	2513	2514	2546	2547	2591	2592	2611	2612	2642	
2643	2663	2664	2695	2696	2723	2724	2755	2756	2781	2782	2815	2816	2844	2845	
2863	2864	2921	2922	2945	2946	2996	2997	3013	3014	3075	3076	3091	3092	3142	
3143	3159	3160	3219	3220	3239	3240	3294	3295	3311	3312	3356	3357	3376	3377	
3431	3432	3449	3450	3500	3501	3528	3529	3572	3573	3645	3646	3668	3669	3683	
3684	3723	3730	3752	3753	3771	3772	3815	3816	3843	3844	3857	3858	3900	3901	
3924	3925	3939	3940	3985	3986	4009	4010	4024	4025	4083	4084	4140	4141	4198	
4199	4254	4255	4310	4311	4354	4355	4381	4382	4507	4503	4517	4518	4593	4594	
4643	4644	4658	4659	4712	4713	4741	4742	4782	4783	4817	4818	4899	4900	4940	
4941	4959	4960	5001	5002	5016	5017	5036	5037	5174	5135	5149	5150	5171	5172	
5260	5261	5277	5278	5321	5322	5355	5356	5398	5399	5421	5422	5449	5450	5522	
5523	5553	5554	5575	5576	5590	5591	5614	5615	5634	5635	5649	5650	5687	5688	
5708	5709	5723	5724	5735	5736	5779	5780	5798	5799	5812	5813	5826	5827	5880	
5881	5922	5923	5939	5940	5980	5981	5995	5996	6007	6008	6039	6040	6071	6072	
6089	6090	6127	6128	6143	6144	6158	6159	6182	6183	6197	6198	6243	6244	6280	
6281	6302	6303	6317	6318	6332	6333	6344	6345	6375	6376	6398	6399	6436	6437	
6454	6455	6995	7279	7280	7281	7345	7346	9056	9057	9294	9295	9296	9297	9544	
9545	10952	10955	10957	10959	10961	10965	10967	10970	10972	10974	10976	10978	10980	10983	
10985	10987	10989	10991	10993	10995	10998	11000	11002	11004	11006	11009	11011	11013	11015	
11018	11020	11022	11024	11027	11029	11031	11036	11038	11041						
.DSABL	9461														
.ENABL		3	9302												
.END	11048														
.ENDC	15	28	30	31	32	36	128	142	154	165	176	401	403	410	423
427	429	457	475	491	492	493	494	498	498	501	523	531	534	537	540
541	542	543	544	547	938	963	964	967	969	971	973	974	975	977	
979	1000	1004	1006	1012	1018	1020	1109	1110	1112	1113	1114	1115	1139	1140	
1142	1143	1144	1145	1168	1169	1203	1204	1205	1206	1408	1409	1413	1414	1415	
1416	1445	1446	1453	1454	1455	1456	1497	1498	1503	1504	1505	1506	1541	1542	
1546	1547	1548	1549	1574	1575	1590	1591	1592	1593	1760	1761	1768	1769	1770	
1771	1807	1808	1816	1817	1818	1819	1855	1856	1866	1867	1868	1869	1926	1927	
1932	1933	1934	1935	1969	1970	1974	1975	1976	1977	2026	2027	2034	2035	2036	
2037	2092	2093	2097	2098	2099	2100	2138	2139	2160	2161	2162	2163	2295	2296	
2302	2303	2304	2305	2414	2415	2428	2430	2430	2431	2495	2496	2503	2504	2505	
2506	2574	2575	2581	2582	2583	2584	2626	2627	2632	2633	2634	2635	2677	2678	
2685	2686	2687	2688	2737	2738	2744	2745	2746	2747	2795	2796	2802	2803	2804	
2805	2890	2891	2898	2899	2900	2901	2935	2953	2971	2972	2979	2980	2981	2982	
3009	3022	3039	3040	3047	3048	3049	3050	3085	3100	3118	3119	3126	3127	3128	
3129	3155	3169	3188	3189	3196	3197	3198	3199	3233	3249	3269	3270	3277	3278	
3279	3280	3307	3321	3340	3341	3346	3347	3348	3349	3413	3414	3421	3422	3423	
3424	3444	3461	3485	3486	3490	3491	3492	3493	3513	3514	3518	3519	3520	3521	
3541	3542	3553	3554	3555	3556	3629	3630	3635	3636	3637	3638	3659	3678	3693	
3712	3713	3719	3720	3721	3722	3743	3766	3781	3799	3800	3805	3806	3807	3808	
3829	3853	3883	3884	3890	3891	3892	3893	3914	3934	3949	3968	3969	3975	3976	
3977	3978	4000	4019	4034	4053	4054	4062	4063	4064	4065	4093	4112	4113	4121	
4122	4123	4124	4150	4169	4170	4178	4179	4180	4181	4208	4226	4227	4235	4236	
4237	4238	4264	4282	4283	4291	4292	4293	4294	4320	4338	4339	4344	4345	4346	
4347	4364	4365	4366	4371	4372	4373	4374	4399	4400	4406	4407	4408	4409	4429	
4430	4478	4479	4484	4485	4486	4487	4512	4527	4546	4547	4557	4558	4559	4560	
4603	4617	4618	4624	4625	4626	4627	4653	4663	4690	4691	4702	4703	4704	4705	
4727	4728	4792	4849	4850	4855	4856	4857	4858	4880	4881	4911	4924	4925	4930	
4931	4932	4933	4954	4969	4970	4971	4986	4987	4988	4989	5011	5026	5103	5104	
5119	5120	5121	5122	5144	5159	5238	5239	5246	5247	5248	5249	5270	5287	5291	
5292	5308	5309	5310	5311	5333	5334	5365	5383	5407	5408	5434	5459	5496	5497	
5510	5511	5512	5513	5548	5563	5564	5569	5570	5584	5585	5610	5623	5624	5628	

5629	5643	5644	5667	5670	5671	5677	5678	5679	5680	5696	5697	5702	5703	5717	
5718	5761	5762	5768	5769	5770	5771	5792	5793	5864	5865	5870	5871	5872	5873	
5899	5900	5908	5909	5910	5911	5957	5961	5962	5969	5970	5971	5972	6047	6048	
6052	6053	6061	6062	6063	6064	6080	6081	6106	6107	6108	6109	6117	6118	6119	
6120	6152	6153	6168	6169	6176	6177	6191	6192	6212	6213	6226	6227	6233	6234	
6258	6259	6270	6271	6272	6273	6289	6290	6296	6297	6311	6312	6326	6327	6370	
6410	6411	6445	6446	6463	6464	6466	6467	6476	6477	6478	6479	6511	6512	6516	
6517	6518	6519	6544	6545	6565	6566	6571	6572	6573	6574	6598	6599	6618	6619	
6631	6632	6633	6634	6665	6666	6678	6679	6782	6783	6795	6796	6797	6798	6829	
6830	6848	6849	6874	6875	6964	6965	6966	6968	6971	6977	6980	6981	6985	6987	
6993	6995	6996	6999	7002	7025	7032	7076	7079	7082	7087	7092	7097	7099	7110	
7113	7125	7127	7129	7136	7140	7145	7147	7151	7154	7155	7228	7233	7234	7237	
7264	7279	7290	7293	7303	7310	7338	7339	7340	7348	7359	7363	7364	7365	7374	
7480	7530	8608	8621	8649	9037	9064	9065	9079	9108	9158	9225	9302	9333	9342	
9346	9377	9379	9394	9425	9461	9465	9476	9488	9489	9497	9499	9502	9530	9547	
9548	9554	9558	9564	9608	9611	9656	9665	9672	9678	9679	9689	9696	9699	9699	
9705	9708	9727	9728	9729	9730	9731	9732	9733	9734	9735	9736	9737	9738	9739	
9740	9741	9742	9743	9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754	
9755	9756	9757	9758	9759	9760										
.EQUIV	36	37	45	60	61	90	91	92	93	94	95	96	97	98	99
.EVEN	118	119	120	121	122	123	124	125	126	127					
.IF	501	1020	7282	7336	9308	9554	9695	10930							
	11	27	29	30	31	32	34	100	128	154	165	176	400	402	409
	422	426	428	457	475	491	492	493	497	498	500	523	531	534	537
	540	541	542	543	544	545	547	938	958	963	965	967	969	971	973
	974	975	977	995	1003	1004	1005	1007	1010	1019	1108	1110	1112	1114	1115
	1138	1140	1142	1144	1145	1167	1169	1203	1205	1206	1407	1409	1413	1415	1416
	1444	1446	1453	1455	1456	1496	1498	1503	1505	1506	1540	1542	1546	1548	1549
	1573	1575	1590	1592	1593	1759	1761	1768	1770	1771	1806	1808	1816	1818	1819
	1854	1856	1866	1868	1869	1925	1927	1932	1934	1935	1968	1970	1974	1976	1977
	2025	2027	2034	2036	2037	2091	2093	2097	2099	2100	2137	2139	2160	2162	2163
	2294	2296	2302	2304	2305	2413	2415	2428	2430	2431	2494	2496	2503	2505	2506
	2573	2575	2581	2583	2584	2625	2627	2632	2634	2635	2676	2678	2685	2687	2688
	2736	2738	2744	2746	2747	2794	2796	2802	2804	2805	2889	2891	2898	2900	2901
	2934	2935	2952	2953	2970	2972	2979	2981	2982	3008	3009	3021	3022	3038	3040
	3047	3049	3050	3084	3085	3099	3100	3117	3119	3126	3128	3129	3154	3155	3168
	3169	3187	3189	3196	3198	3199	3232	3233	3248	3249	3268	3270	3277	3279	3280
	3306	3307	3320	3321	3339	3341	3346	3348	3349	3412	3414	3421	3423	3424	3443
	3444	3460	3461	3484	3486	3490	3492	3493	3512	3514	3518	3520	3521	3540	3542
	3553	3555	3556	3628	3630	3635	3637	3638	3658	3659	3677	3678	3692	3693	3711
	3713	3719	3721	3722	3742	3743	3765	3766	3780	3781	3798	3800	3805	3807	3808
	3828	3829	3852	3853	3882	3884	3890	3892	3893	3913	3914	3933	3934	3948	3949
	3967	3969	3975	3977	3978	3999	4000	4018	4019	4033	4034	4052	4054	4062	4064
	4065	4092	4093	4111	4113	4121	4123	4124	4149	4150	4168	4170	4178	4180	4181
	4207	4208	4225	4227	4235	4237	4238	4263	4264	4281	4283	4291	4293	4294	4319
	4320	4337	4339	4344	4346	4347	4363	4364	4366	4371	4373	4374	4398	4400	4406
	4408	4409	4429	4477	4479	4484	4486	4487	4511	4512	4526	4527	4545	4547	4557
	4559	4560	4602	4603	4616	4618	4624	4626	4627	4652	4653	4667	4668	4689	4691
	4702	4704	4705	4727	4791	4792	4848	4850	4855	4857	4858	4880	4910	4911	4923
	4925	4930	4932	4933	4953	4954	4968	4969	4971	4986	4988	4989	5010	5011	5025
	5026	5102	5104	5119	5121	5122	5143	5144	5158	5159	5237	5239	5246	5248	5249
	5269	5270	5286	5287	5290	5292	5308	5310	5311	5333	5364	5365	5388	5389	5407
	5433	5434	5458	5459	5495	5497	5510	5512	5513	5547	5548	5563	5569	5584	5609
	5610	5623	5628	5643	5666	5667	5669	5671	5677	5679	5680	5696	5702	5717	5760
	5762	5768	5770	5771	5792	5863	5865	5870	5872	5873	5898	5900	5908	5910	5911
	5956	5957	5960	5962	5969	5971	5972	6047	6051	6053	6061	6063	6080	6106	