RL11/RLV11,RL01

RL01 DRIVE TEST PART 2

MD-11-DZRLD-A

EP-DZRLD-A-DL

FICHE 1 OF 1

DEC 1977

digital

MADE IN USA

IDENTIFICATION

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

PRODUCT NAME: RL01 DRIVE TEST (PART 2)

DATE CREATED: 11 OCTOBER 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: M. TEGROTENHUIS/D. DEKNIS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH BOTH XXDP AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP, AND CAN BE CHAINED UNDER XXDP, ACT AND APT IN ACT MODE (SEE "CREATE CORE IMAGE" COMMAND BELOW FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT WE HAVE INCORPORATED INTO IT A CONTROL MODULE WHICH WILL LATER BE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DS A>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED BELOW.

THE SUPERVISOR CODING FOLLOWS IMMEDIATELY THE DIAGNOSTIC TEST CODING, BUT THE SUPERVISOR LISTING HAS BEEN SUPPRESSED FOR GENERAL DISTRIBUTION. A LIMITED DISTRIBUTION HAS BEEN MADE TO FIELD SERVICE OF THE SUPERVISOR ASSEMBLY LISTING, AND IT MAY BE CONSULTED IN EVENT OF A SOFTWARE PROBLEM.

1.1.2 DIAGNOSTIC INFORMATION

THIS PROGRAM TESTS AND EXERCISES RL01 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM FIRST TESTS THE RL01 INTERFACE AND BASIC DRIVE LOGIC. IT THEN BEGINS TESTING THE SEEK OPERATIONS USING SINGLE DIFFERENCES, PROCEEDING INTO SEEKS OF GREATER DIFFERENCES. SEEK TIMING IS DONE AFTER THE SEEK LOGIC HAS BEEN TESTED.

DATA TRANSFERS ARE DONE AFTER ALL THE SEEK TESTS. THE FIRST DATA TRANSFER IS READING OF THE BAD SECTOR FILES WHICH ARE STORED AND USED LATER TO PREVENT TESTING ON BAD SECTORS. FOLLOWING DATA READ AND WRITE TESTING, THE PROGRAM TESTS FOR OVERWRITE PROBLEMS AND ADJACENT CYLINDER INTERFERENCE.

SEEK TIMING, ROTATIONAL TIMING, AND WRITE LOCK DATA PROTECTION ARE DONE IF MANUAL INTERVENTION IS REQUESTED.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF CORE

CONSOLE DEVICE (LA30, LA36, VT50, ETC.)
RL11/RLV11 CONTROLLER(S)
1 - 8 RL01 DRIVES
1 - 8 RL01K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

MAINDEC-11-DZRLD-A

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RL01-UG-PRE)
XXDP USERS MANUAL
DIAGNOSTIC SUPERVISOR PROGRAM LISTING

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

MD-11-DZRLA	RL11/RLV11 RL01 CONTROLLER TEST (PART 1)
MD-11-DZRLB	RL11/RLV11 RL01 CONTROLLER TEST (PART 2)
MD-11-DVRLA	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
MD-11-DZRLC	RL01 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES

FOLLOW STANDARD DEC PROCEDURES TO LOAD THE PROGRAM. (XXDP, ABSOLUTE LOADER, UPD1, UPD2)

2.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

2.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE WITHOUT READING THE REMAINDER OF THIS DOCUMENT AS FOLLOWS:

A) LOAD THE DIAGNOSTIC
B) START AT ADDRESS 200
C) ANSWER THE HARDCORE QUESTIONS
D) RECEIVE PROMPT (DS A>)
E) ENTER STA<CR>
F) ANSWER HARDWARE AND SOFTWARE QUESTIONS
G) GET END OF PASS MESSAGES OR ERROR MESSAGES
H) TO END EXECUTION, ENTER CONTROL/C

2.2 SPECIAL ENVIRONMENTS

THE ENVIRONMENTS THIS PROGRAM WILL RUN IN ARE XXDP, XXDP CHAIN, ACT, SLIDE AND APT.

2.3 PROGRAM OPTIONS

2.3.1 START COMMAND

```
**********************************************************************
STA(RT)/TESTS: <TEST-LIST>/PASS: <PASS-CNT>/FLAGS: <FLAG-LIST>/EOP: <INCR>
**********************************************************************
```

2.3.1.1 TESTS SWITCH (/TESTS: <TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1: 2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5: 8-10 ETC.) SEPARATED BY COLONS, SPECIFYING WHICH TESTS IT IS DESIRED BE EXECUTED. THE TEST NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 2.3.1.

2.3.1.2 PASS SWITCH (/PASS: <PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION: IE, EXIT IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY A HALT ON ERROR BEING ENCOUNTERED, IN WHICH CASE WE RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 2.3.1.

2.3.1.3 FLAGS SWITCH (/FLAGS: <FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDR	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 2.3.1.

2.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 2.3.1.

2.3.1.5 EFFECT OF COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWHRE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 64. THE TERM "UNIT" REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

AT THE POINT WHERE THE QUESTION "# UNITS?" IS ANSWERED, CORE STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS: 1: 2-4: 6: 8-10/PASS: 3/FLAGS: IER: HOE=1: UAM: LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

2.3.2 RESTART COMMAND

****
RES(TART)/TESTS: <TEST-LIST>/PASS: <PASS-CNT>/FLAGS: <FLAG-LIST>/UNITS: <UNIT-LIST>
****

2.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

2.3.2.2 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1,2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5, 8-10 ETC.) SEPARATED BY COLONS, INDICATING WHICH UNITS IT IS DESIRED BE TESTED. THE NUMBERS MAY RANGE FROM 1 THRU N (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2.3.2.3 EFFECT OF COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2.3.3 CONTINUE COMMAND

****
CON(TINUE)/PASS: <PASS-CNT/FLAGS: <FLAG-LIST>
****

2.3.3.1 PASS SWITCH (/PASS: <PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

2.3.3.2 FLAG SWITCH (/FLAGS: <FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RE-

TAIN THEIR CURRENT VALUE.

2.3.3.3 EFFECT OF COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

2.3.4 PROCEED COMMAND

****
PRO(CEED)/FLAGS: <FLAG-LIST>
****

2.3.4.1 FLAGS SWITCH (/FLAGS: <FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2.3.4.2 EFFECT OF COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

2.3.5 CREATE CORE IMAGE COMMAND

****
CCI/TESTS: <TEST-LIST>/PASS: <PASS-CNT>/FLAGS: <FLAG-LIST>
****

2.3.5.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, <FLAG-LIST>, AND ARE AS IN THE START COMMAND, EXCEPT THAT THE UAM (UNATTENDED MODE) FLAG DEFAULTS TO THE SET POSITION.

2.3.5.2 EFFECT OF COMMAND

THE PURPOSE OF THIS COMMAND IS TO CREATE A BIC FILE SUITABLE FOR CHAIN MODE EXECUTION. THE XXDP PROCEDURE IS AS FOLLOWS:

```
INVOKE THE XXDP UTILITY UPD1
LOAD XXN:FILE.BIN
START 200
<QUESTIONS AND ANSWERS>
RESTART UPD1 USING RESTART ADDRESS
HICORE ADDRESS (IF "PASSED 14.5K" MESSAGE CAME)
DUMP XXN:FILE.BIC
```

THE OPERATOR DIALOGUE (HARDWARE AND SOFTWARE) WILL BE EXECUTED AS IN THE START COMMAND, BUT AT THE END OF THE QUESTIONS THE HALT STATE WILL BE ENTERED, AT WHICH TIME THE OPERATOR SHOULD

DUMP THE PROGRAM TO THE XXDP LIBRARY USING A BIC EXTENSION
TO INDICATE THAT THIS FILE IS CHAINABLE. HE SHOULD USE THE XXDP UTILITY "UPD1" TO DO THIS. IF THE P-TABLES EXTEND BEYOND 14.5K, A MESSAGE WILL BE ISSUED GIVING THE NEW UPPER CORE LIMIT, TO WHICH THE OPERATOR MUST ADJUST BEFORE DUMPING. HE MAY NOW DELETE THE NON-CHAINABLE BIN FILE IF DESIRED, SINCE THE BIC FILE HAS ALL THE CAPABILITIES OF IT.

WHEN THIS BIC FILE IS SUBSEQUENTLY EXECUTED IN CHAIN MODE, THE OPERATOR DIALOGUES WILL BE BYPASSED. HOWEVER, IF IT IS EXECUTED STANDALONE, THE DIALOGUE WILL BE REISSUED.

NOTE THAT IF THE MESSAGE "TOO MANY UNITS" IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

2.3.6 ADD COMMAND

****
ADD/UNITS: <UNIT-LIST>
****

2.3.6.1 UNITS SWITCH (/UNITS: <UNIT-LIST>

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.6.2 EFFECT OF COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

2.3.7 DROP COMMAND

****
DRO(P)/UNITS: <UNIT-LIST>
****

2.3.7.1 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.7.2 EFFECT OF COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

2.3.8 PRINT COMMAND

****
PRI(NT)
****

2.3.8.1 EFFECT OF COMMAND

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

SEQ 0009

2.3.9 DISPLAY COMMAND

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

2.3.9.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.9.2 EFFECT OF COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

2.3.10 FLAGS COMMAND

****
FLA(GS)
****

2.3.10.1 EFFECT OF COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.3.11 ZFLAGS COMMAND

****
ZFL(AGS)
****

2.3.11.1 EFFECT OF COMMAND

ALL FLAGS ARE CLEARED.

2.3.12 CONTROL CHARACTERS

A CONTROL C (C) ENTERED VIA THE CONSOLE DEVICE DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO THE DIAGNOSTIC SUPERVISOR COMMAND MODE.

A CONTROL Z (Z) ENTERED WITHIN ONE OF THE THREE OPERATOR DIALOGUES(HARDCORE, HARDWARE, OR SOFTWARE QUESTIONS) CAUSES THE DEFAULT VALUES TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL CONSOLE DEVICE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER CONTROL O IS TYPED.

2.3.13 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONCE.

RL11 (L) Y?

ANSWER YES(Y) IF YOU HAVE AN RL11 CONTROLLER, NO(N) IF YOU HAVE AN RLV11 CONTROLLER.

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER.

2.3.14 SOFTWARE PARAMETERS

AFTER HARDWARE PARAMETERS ARE SPECIFIED, THE USER IS ASKED IF SOFTWARE PARAMETERS ARE TO BE CHANGED.

CHANGE S.W. (Y OR N)? (NO DEFAULT)

IF "NO", SOFTWARE PARAMETER ENTRY IS SKIPPED, ALL DEFAULT VALUES ARE USED. IF "YES", THE PARAMETERS LISTED BELOW ARE REQUESTED. TYPING CONTROL Z IN RESPONSE TO ANY PARAMETER QUESTION ALLOWS THIS AND ALL REMAINING PARAMETERS TO DEFAULT.

USE ALL CYLINDERS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SELECTED SET OF CYLINDERS WILL TEST EVERY CYLINDER ON THE CARTRIDGE.

USE ALL SECTORS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SINGLE SECTOR TO TEST A GIVEN OPERATION (SUCH AS SEEK DESTINATION) WILL READ AND VERIFY EVERY SECTOR HEADER.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF "YES", SEEK TIMING, ROTATIONAL TIMING, AND WRITE LOCK ERROR AND DATA PROTECTION TESTS ARE EXECUTED. THE ONLY TEST THAT ACTUALLY REQUIRES MANUAL INTERVENTION IS THE WRITE LOCK TEST AND THAT TEST WILL BYPASS AUTOMATICALLY AFTER WAITING 30

SECONDS FOR WRITE LOCK TO BE SET.

LOWER SEEK LIMIT (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

THIS LIMIT IS IMPOSED ON ALL SEEK OPERATIONS SUCH THAT TESTING IS NOT DONE BELOW THAT LIMIT. IN ADDITION, SETTING THIS LIMIT (OR THE UPPER LIMIT, SEE BELOW) CAUSES THE FORWORD AND REVERSE OSCILLATING SEEK TESTS TO PERFORM DIFFERENTLY (SEE TEST DESCRIPTION). TESTS THAT REQUIRE ACCESS TO A SPECIFIC CYLINDER THAT FALLS BELOW THE SPECIFIED LIMIT WILL IGNORE THE LIMIT (SEE WRITE/READ TEST PART 1).

UPPER SEEK LIMIT (N)?

IF "YES", AN UPPER CYLINDER LIMIT IS IMPOSED IN THE SAME MANNER AS THE LOWER SEEK LIMIT. A "YES" RESPONSE WILL CAUSE THE FOLLOWING PARAMETER REQUEST.

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

SPECIFY SURFACE (0 OR 1) (DECIMAL) (0)?

WHICHEVER SURFACE IS SPECIFIED IS THE ONLY SURFACE TESTED IN THE ENTIRE PROGRAM. ANY TEST THAT IS DESIGNED TO TEST THE OTHER SURFACE IS AUTOMATICALLY BYPASSED. THE PROGRAM DOES NOT PRINT ANY INDICATION THAT A TEST IS BYPASSED IN THIS CASE.

SPECIFY ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE MAXIMUM NUMBER OF ERRORS ALLOWED. THIS LIMIT IS ON A PER DRIVE BASIS IN A SINGLE PASS. IF THE ERROR LIMIT IS EXCEEDED, THE DRIVE IS DROPPED FROM FURTHUR TESTING.

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE NUMBER OF DATA COMPARE ERRORS THAT WILL BE LISTED FOR A GIVEN COMPARE OPERATION. AFTER THE LIMIT IS REACHED, THE DATA ERRORS ARE NOT PRINTED BUT THE COMPARE CONTINUES UNTIL THE END OF THE DATA FIELD. A TOTAL IS REPORTED AT THE END OF THE COMPARE.

DROP DRIVE IF NO RESPONSE (N)?

IF THIS PARAMETER IS SPECIFIED AS YES, THE PROGRAM WILL CHECK

IF THE DRIVE IS READY OR IF IT WILL RESPOND TO A GET STATUS BEFORE TESTING STARTS ON THAT DRIVE. IF IT IS NOT READY AND WILL NOT RESPOND TO A GET STATUS THE DRIVE IS DROPPED AND A MESSAGE IS PRINTED.

2.3.15 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED THEN AND THERE TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D)  ?  64

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 1-20
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 21-49,,51-64
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL
64 TABLES. SLOT TWO RECEIVES THE VALUES 1,2,3,...,20 IN TABLES 1 THRU 20 SEQ 0013
AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76
IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 THRU THE END ARE GOING TO
BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE
THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES).
QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS
AT CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXLICIT VALUES ARE TYPED
IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND
GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES
51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64
EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
(NAMELY QUESTION 2).

2.4 EXECUTION TIMES

FIRST PASS EXECUTION ON 1 DRIVE WITH MANUAL INTERVENTION WILL
REQUIRE APPROXIMATELY 8 MINUTES. SECOND AND SUBSEQUENT PASS
RUN TIME IS APPROXIMATELY 20 MINUTES.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

3.1.1 SPECIFIC OPERATION MESSAGES

THE OPERATION MESSAGE (LINE 4) IS GENERATED IN A DYNAMIC
MANNER BASED ON THE SUBSYSTEM FUNCTION BEING EXECUTED AT THE
TIME OF THE ERROR AND THE STATE OF THE FLAGS IN THE LOCATION
TAGGED "OPFLAGS". THE POSSIBLE OPERATION MESSAGES ARE GIVEN
BELOW.

SEEK -
FROM (CYL NUM) DIFF (CYL DIFF) SGN (0 OR 1) HD (0 OR 1) WHERE
THE VALUES ARE GIVEN IN OCTAL. THIS MESSAGE IS THE RESULT OF
A SEEK OPERATION THAT WAS VERIFIED BY A READ HEADER AND THE
HEAD POSITION AFTER A SEEK IS IN ERROR. (THE ACTUAL HEAD
POSITION IN THIS ERROR SITUATION IS GIVEN IN THE RESULT LINE,
LINE 5.)

READ DATA -
IS A READ DATA OPERATION WHERE SOME FORM OF ERROR WAS DETECTED
IN THE ACTUAL READ OPERATION. THIS ERROR COULD BE HARDWARE
DETECTED SUCH AS DATA CRC, HEADER CRC, HEADER NOT FOUND, ETC.,
OR A SOFTWARE DETECTED ERROR SUCH AS DRIVE READY RESET AFTER A
READ DATA COMPLETED.

READ DATA WITH DATA COMPARE -
IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER

A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT
INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO

DETECTED ERRORS BUT THE DATA WAS WRONG.

READ HEADER -
READ HEADER FOR 40 HEADERS -
READ HEADER FOR 40 HEADERS WITH HEADER COMPARE -
HAVE THE SAME GENERAL MEANING AS THE READ DATA AND READ DATA WITH DATA COMPARE. MESSAGES HAVING THE OPERATION OF READ HEADER OR READ HEADER FOR 40 HEADERS ARE THE RESULT OF ERRORS DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

WRITE DATA -
RESET -
GET STATUS -
GET STATUS WITH RESET -
ARE ALL BASIC OPERATIONS. AS BEFORE, THE ERROR DETECTION CAN BE EITHER HARDWARE OR SOFTWARE. THE RESULT LINE (LINE 5) WILL DEFINE THE REASON FOR THE REPORT.

LD DRV -
UNLD DRV -
ARE OPERATION MESSAGES THAT WILL APPEAR IN THE REPORT WHEN THE DRIVE LOAD AND UNLOAD SEQUENCE IS BEING TESTED.

ANOTHER GROUP OF OPERATION QUALIFIERS WILL BE REPORTED FOR OPERATIONS THAT FAIL IN SPECIFIC TESTS. THESE TESTS ARE THE WRITE/READ TEST PART 2, OVERWRITE TEST, AND THE ADJACENT CYLINDER INTERFERENCE TEST.

OPERATION	QUALIFIER
READ DATA WITH DATA COMPARE	FOL 0 TO CC SEEK
READ DATA	FOL 255 TO CC SEEK
WRITE DATA	FOL WRITE (NO SEEK)
READ HEADER	ADJ. CYL WRITTEN AFTER FWD SK
	ADJ. CYL WRITTEN AFTER REV SK
	SK FWD, WRT-SK REV, OVERWRT
	SK REV, WRT-SK FWD, OVERWRT

THE ABOVE OPERATIONS CAN BE REPORTED WITH ANY OF THE QUALIFIERS. THE QUALIFIERS IN THESE TESTS ARE AN ATTEMPT TO MAKE THE REPORT MORE MEANINGFUL BY PROVIDING INFORMATION ABOUT THE SEQUENCE OF OPERATIONS BEING DONE.

THE QUALIFIERS "FOL 0 TO CC SEEK" AND "FOL 255 TO CC SEEK" INDICATE THAT THE SEQUENCE OF OPERATIONS INCLUDED A SEEK OF A GIVEN DIRECTION TO THE CYLINDER WHERE THE TEST IS BEING

PERFORMED.

THE "FOL WRITE (NO SEEK)" QUALIFIER MEANS THAT THE OPERATION WAS DONE AFTER A WRITE WITH NO HEAD MOVEMENT BETWEEN THE WRITE AND READ.

THE QUALIFIER "ADJ CYL WRITTEN AFTER FWD SK" AND "ADJ CYL WRITTEN AFTER REV SK" WILL BE REPORTED ONLY IN THE ADJACENT CYLINDER INTERFERENCE TEST. THESE QUALIFIERS ARE USED WHEN THE ERROR OCCURS ON THE CYLINDER UNDER TEST AND DEFINE THE

DIRECTION THE HEADS WERE MOVED WHEN THE ADJACENT CYLINDER WAS WRITTEN.

THE QUALIFIERS "SK FWD, WRT-SK REV, OVERWRT" AND "SK REV, WRT-SK FWD, OVERWRT" WILL BE REPORTED ONLY IN THE OVERWRITE TEST. THESE QUALIFIERS DEFINE THE DIRECTION OF HEAD MOTION BEFORE THE INITIAL WRITE AND THE OVERWRITE.

THE QUALIFIER "ON BAD SEC FILES" WILL BE REPORTED WITH THE WRITE DATA COMMAND IF THE PROGRAM ABORTS THAT COMMAND BECAUSE THE WRITE WOULD BE ON THE BAD SECTOR FILES.

3.1.2 SPECIFIC RESULT MESSAGES

THE RESULT MESSAGE (LINE 5) IS GENERATED DYNAMICALLY BASED ON THE EXPECTED RESULT OF THE OPERATION BEING TESTED. SINCE OPERATIONS ARE MONITORED DURING EXECUTION THE RESULT MESSAGE MAY REPORT AN ERROR DETECTED DURING THE OPERATION AS WELL AS THE ERRORS SEEN AT THE END OF THE OPERATION. ONLY THE FIRST ERROR SEEN IS REPORTED IN ALL CASES.

THE GENERAL FORMAT FOR THE RESULT LINE IS

RESULT: (VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)

WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUCH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HEADER CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HEADER NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT THE RESULT

ACTUALLY IS. THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT CONDITIONS, A NUMBER 0 TO 7 TO INDICATE THE DRIVE STATE, OR A NUMBER 0 TO 377 (OCTAL) TO IDENTIFY A CYLINDER NUMBER.

VARIABLE 3 DEFINES THAT THE VALUE GIVEN IS VARIABLE 2 SHOULD BE.

THE OPTIONAL QUALIFIER IS PROVIDED WHEN IT IS USEFUL TO KNOW WHEN THE ERROR WAS DETECTED IN THE OPERATION BEING PERFORMED. THIS QUALIFIER IS USED TO REPORT RESULTS SUCH AS:

```
BRUSH HME IS 1 SB 0 IN STATE 2
HEADS OUT IS 0 SB 1 IN STATE 3
DRV RDY   IS 0 SB 1 IN DATA XFER
SELECTED HEAD IS 1 SB 0 IN CYCLE UP
DRV RDY   IS 0 SB 1 IN STATE 5
DRV RDY   IS 1 SB 0 IN SEEK W/O MOTION
DRV RDY   IS 0 SB 1 IN 10MS
DRV RDY   IS 0 SB 1 IN 500MS
DRV RDY   IS 0 SB 1 IN 5SECONDS
```

THESE RESULTS, WHEN SEEN WITH THE OPERATION MESSAGE, WILL BE SELF EXPLANATORY.

OTHER RESULT MESSAGES THAT CAN BE PART OF AN ERROR REPORT ARE:

"INTERRUPT TO LATE"
WHICH INDICATES THAT THE OPERATION BEING PERFORMED DID NOT COMPLETE IN THE EXPECTED AMOUNT OF TIME. THIS RESULT CAN BE CAUSED BY THE DRIVE LOOSING READY BEFORE STARTING A READ HEADER AND THEREFORE NOT COMPLETING THE READ HEADER IN 1MS.

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"
IS REPORTED WHEN AN ERROR CAUSES HEADS TO UNLOAD AND AFTER THE ERROR IS CLEARED THE HEADS DO NOT RELOAD.

"UNKN DRV STATE-NO RDY, NO ERR, HDS OUT"
IS REPORTED WHEN THE PROGRAM CANNOT DETERMINE THE DRIVE STATE OR STATUS.

"WRITE ABORTED"
IS REPORTED WHEN THE PROGRAM ABORTS A WRITE TO PROTECT THE BAD SECTOR FILES.

"COULD NOT RETRIEVE DRIVE STATUS"
IS REPORTED IF THE GET STATUS COMMAND DOES NOT COMPLETE SUCCESSFULLY WHEN THE STATUS IS REQUIRED TO REPORT AN ERROR.

"OPI SET-NO DRIVE RESPONSE"
IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (OPI SETS) WHEN THAT COMMAND IS BEING USED IN THE EARLY TESTS TO CHECK THE DRIVE INTERFACE.

"NO INTERRUPT ON CMND COMPLETE"
IS REPORTED WHEN THE COMMAND SUCCESSFULLY COMPLETES BUT THE CONTROLLER HAS NOT GENERATED AN INTERRUPT.

"ERR DID NOT CLEAR" IS REPORTED WHEN THE RESET COMMAND DOES NOT CLEAR THE CONTROLLER ERRORS. THIS IS A CONTROLER RELATED

PROBLEM BUT IS REPORTED IF SEEN IN THE DRIVE TEST PROGRAMS.

"DRV ERR IS NOT CLEARED"
IS REPORTED WHEN THE GET STATUS W/RESET COMMAND DOES NOT CLEAR ALL DRIVE ERRORS.

"UNEXPECTED ERR"
IS REPORTED WHEN THE CONTROLLER SENSES AN ERROR BUT NO ERROR BITS ARE SET.

"BAD SEC FILE FMT ERR"
IS REPORTED IF THE CONTENTS OF THE FILES DO NO CORRESPOND TO THE EXPECTED FORMAT (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

"BAD SEC FILES NOT STRD. ALL SEC ASSUMED GOOD."
THIS MESSAGE IS PRINTED WHEN A PARTICULAR TEST REQUIRES THE BAD SECTOR FILES BUT THEY HAVE NOT BEEN STORED. THIS SITUATION WILL OCCUR IF THIS TEST IS STARTED OUT OF THE NORMAL PROGRAM SEQUENCE OR IF THE BAD SECTOR FILES COULD NOT BE READ.

"ERROR LIMIT EXCEEDED-UNIT DROPPED"
IS REPORTED (WITH THE UNIT NUMBER) WHEN MORE THAN THE

SPECIFIED NUMBER OF ERRORS (DEFAULT 20) HAVE OCCURED IN ANY SINGLE PASS.

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

```
(1)   PROG NAME   ERR NUM   TEST NUM   SUBTEST NUM   ERR PC
(2)   ROUTINE TRACE SEQ (IN SEQ CALLED)
         (ADDRESS)
         (ADDRESS)
            .
            .
         (ADDRESS)
(3)   TEST DESCRIPTION
(4)   OPERATION:
(5)   RESULT:
(6)   ADDRESS OF UNIT UNDER TEST
(7)                RLCS    RLDA    RLBA    RLMP    CYL     HD
(8)   OP INIT
(9)   OP DONE
(10)        DRIVE STATUS
(11)              WORD NUM IS (XXXXXX) SB (YYYYYY)
(12)   TOTAL COMPARE ERRS: (ZZZ) OF (128)
```

THE ONLY EXCEPTION TO THE ABOVE FORMAT IS PURE DATA COMPARE ERRORS (NOT DETECTED BY READ ERROR). THEN THE FORMAT DOES NOT INCLUDE LINES 5 THROUGH 10.

LINE 1 IS THE ERROR HEADER AND IS PROVIDED BY THE SUPERVISOR. THE PROGRAM IS IDENTIFIED BY NAME WITH THE NUMBER OF TEST AND SUBTEST PRESENTLY BEING EXECUTED.

THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT IT DOES

NOT REFER TO A PHYSICAL SUBTEST WITHIN A GIVEN TEST. RATHER IT REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED WITHIN A TEST. CONSEQUENTLY, ON A TEST THAT TESTS AN INCREMENTAL TYPE OF OPERATION (SUCH A INCREMENTAL SEEKS, READ ALL HEADERS FROM BOTH SURFACES, ETC.) THE SUBTEST WILL BE DESCRIPTIVE OF WHERE IN THE TEST THE ERROR OCCURRED.

THE ERROR P.C. IS THE PHYSICAL MEMORY LOCATION WHERE THE ERROR REPORT WAS INITIATED. SINCE MANY FUNCTIONS ARE SUBROUTINED, AND ERRORS ARE REPORTED FROM SUBROUTINES, THE ERROR P.C. IS NOT SUFFICIENT TO IDENTIFY THE LOCATION OF THE ERROR CALL AND THE ROUTINE TRACE SEQUENCE IS PROVIDED.

LINE 2 IS THE ROUTINE TRACE SEQUENCE. IF THE ERROR CALL IS INITIATED FROM WITHIN THE TEST (AS OPPOSED TO WITHIN A ROUTINE), THIS PORTION OF THE REPORT IS OMITTED. IF THE CALL IS INITIATED FROM A ROUTINE (WHICH MAY BE CALLED BY ANOTHER ROUTINE, WHICH MAY BE CALLED BY ANOTHER ROUTINE, ETC. SEVERAL LEVELS DEEP) THE ROUTINE TRACE SEQUENCE PROVIDES A TRAIL TO THE ACTUAL LOCATION WITHIN THE TEST THAT CALLED THE FIRST ROUTINE. THE FIRST ENTRY LISTED IS THE LOCATION WHERE THE FIRST ROUTINE WAS CALLED.

LINE 3 IS THE TEST DESCRIPTION AND IS ROUGHLY IDENTICAL TO THE NAME OF THE TEST BEING PERFORMED.

LINE 4 IDENTIFIES THE ACTUAL HARDWARE FUNCTION THAT IS BEING PERFORMED. ADDITIONAL INFORMATION ON THIS LINE IS DESCRIPTIVE OF SPECIFIC USE OF THE FUNCTION. FOR EXAMPLE, THE OPERATION LINE WILL READ "READ HEADERS FOR 40 HEADERS" WHEN ALL HEADERS ARE BEING READ FROM A TRACK.

LINE 5 IDENTIFIES THE ERROR THAT HAS BEEN DETECTED. THE CONTENT OF LINE 5 IDENTIFIES WHAT WAS BENG TESTED (SUCH AS DRIVE READY, CONTROLLER ERROR, DRIVE STATE, ETC.), WHAT IT IS AND WHAT IT SHOULD BE. LINE 5 MAY BE REPEATED IF MORE THAN ONE TESTED ITEM IS FOUND IN ERROR.

IN ADDITION LINE 5 WILL REPORT ANY HARDWARE DETECTED ERRORS SUCH AS OPERATION INCOMPLETE, HEADER CRC, ETC. IN THIS CASE THE FIRST LINE PRINTED AS RESULT WILL BE DETERMINED BY THE THREE ERROR BITS OPI, HNF/DLT, AND HCRC/DCRC. THE LINE WILL BE DETERMINED AS IN THE FOLLOWING TRUTH TABLE:

HNF/DLT	DCRC/HCRC	OPI	MESSAGE
1	1	1	HDR NOT FND/HDR CRC/OPI ERROR
0	1	1	HDR CRC ERROR
1	0	1	HDR NOT FND ERROR
0	1	0	DATA CRC ERROR
1	0	0	DATA LATE ERROR

LINE 6 IDENTIFIES THE PHYSICAL ADDRESS OF THE UNIT UNDER TEST. THIS ADDRESS IS BY UNIBUS ADDRESS OF THE CONTROLLER AND DRIVE NUMBER.

LINE 7 NAMES THE CONTROLLER REGISTERS (AND CYLINDER AND HEAD WHERE THESE ARE APPLICABLE IN THE REPORT) TO BE REPORTED.

LINE 8 PROVIDES THE CONTENTES OF CONTROLLER REGISTERS WHEN THE OPERATION WAS INITIATED.

LINE 9 PROVIDES THE CONTENTS OF THE CONTROLLER REGISTERS WHEN THE ERROR BEING REPORTED WAS DETECTED. FREQUENTLY THE REGISTER CONTENTS OF OP INIT AND OP DONE WILL BE DIFFERENT. OP INIT MAY INDICATE A SEEK WAS BEING PERFORMED BUT OP DONE MAY INDICATE THE ERROR WAS DETECTED BY A READ HEADER. THE

REASON IS THAT A SEEK WAS EXECUTED AND DID NOT PROPERLY POSITION HEADS AND WHEN THE READ HEADER WAS DONE THE HEADS WERE ON THE WRONG CYLINDER.

LINE 10 IS THE DRIVE STATUS. THIS LINE IS ONLY REPORTED IF THE RLMP REGISTER DOES NOT CONTAIN THE ACTUAL DRIVE STATUS.

LINE 11 AND LINE 12 ARE REPORTED IF THE ERROR WAS DETECTED AS A COMPARE OPERATION, EITHER DATA OR HEADERS. IN ADDITION, GOOD AND BAD DATA IS REPORTED FOR ALL READ ERRORS.

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR CONTROL OF THE SUBSYSTEM.

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR
BIT 14 - DRIVE ERROR
BIT 13 - NON EXISTANT MEMORY ERROR
BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
       - DATA LATE (WITH BIT 10 CLEAR)
BIT 11 - HEADER CRC (WITH BIT 10 SET)
       - DATA CRC (WITH BIT 10 CLEAR)
BIT 10 - OPERATION INCOMPLETE
BIT 9/8 - DRIVE SELECT (0-3)
BIT 7 - CONTROLLER READY
BIT 6 - INTERRUPT ENABLE
BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
BIT 3-1 - FUNCTION CODE
        0 - NOP (PDP-11) MAINT (LSI-11)
        1 - WRITE CHECK
        2 - GET DRIVE STATUS
        3 - SEEK

4 - READ HEADER
5 - WRITE DATA
6 - READ DATA
7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

BITS 15-1 BUS ADDRESS OF DATA TRANSFER
BIT 0 SHOULD BE 0

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15 - MUST BE ZERO(0)
BIT 14-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (0-47)

FOR SEEK FUNCTION

BIT 15 - MUST BE ZERO(0)
BIT 14-7 - DIFFERENCE TO NEW CYLINDER
BIT 6-5 - MUST BE ZERO(0)
BIT 4 - SURFACE
BIT 3 - MUST BE ZERO
BIT 2 - SEEK DIRECTION(1 - IN / 0 - OUT)
BIT 1 - MUST BE ZERO
BIT 0 - MUST BE ONE(1)

FOR GET STATUS FUNCTION

BIT 15-4 - IGNORED SHOULD BE ZERO
BIT 3 - DRIVE RESET
BIT 2 - MUST BE ZERO
BIT 1 - MUST BE ONE
BIT 0 - MUST BE ONE

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

BIT 15 - 0 - WORD COUNT(TWO'S COMPLIMENT)

FOR READ HEADER FUNCTION

BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
- ZERO WORD (SECOND READ)
- HEADER CRC (THIRD READ)

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR(CHE)
BIT 13 - WRITE LOCK STATUS(WL)
BIT 12 - SEEK TIME OUT(SKTO)
BIT 11 - SPIN ERROR(SPE)
BIT 10 - WRITE GATE ERROR(WGE)
BIT 9 - VOLUME CHECK(VC)
BIT 8 - DRIVE SELECT ERROR(DSE)
BIT 7 - RESERVED(0)
BIT 6 - SURFACE
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 -STATE BITS
0 - LOAD STATE
1 - SPIN UP
2 - BRUSH CYCLE
3 - LOAD HEADS
4 - SEEK - TRACK COUNTING
5 - SEEK - LINEAR MODE
6 - UNLOAD HEADS
7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 DIFFERENCE OF 1 SEEK TEST (PART 1)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.

DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT.

DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 4. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
DIFFERENCE REGISTER DROPPED BIT
STATE ROM FAILED

WAIT APPROX 5 MS. DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 5. IF NOT:

DIFFERENCE REGISTER NOT COUNTING
COUNT PULSE NOT GENERATED (COUNT LOGIC)
SEEK ROM FAILED
FAILURE IN DC SERVO
NO TACH FEEDBACK

WAIT APPROX 5 MS LONGER. TEST DRIVE READY. IF SET:

FAILURE IN READY LATCH OR INTEGRATOR

WAIT APPROX 5 MS LONGER. TEST READY. IF RESET:

FAILURE IN INTEGRATOR
UNEXPECTED GUARD BAND DETECTED

DO SEEK WITH DIFFERENCE 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 2 DIFFERENCE OF 1 SEEK TEST (PART 2)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.

DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT. COMPARE CYLINDER OF THIS HEADER WITH CYLINDER OF STORED HEADER FOR DIFFERENCE OF ONE. IF NOT:

COUNT LOGIC BAD
INTERGRATOR FAILED

CHECK THAT HEADS MOVED FORWARD OR REVERSE AS EXPECTED. IF NOT:

SEEK ROM FAILED

DO SEEK WITH DIFFERENCE OF 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 3 OUTER GUARD BAND DETECTION TEST

DO READ HEADER, WAIT FOR INTERRUPT. CHECK IF AT CYLINDER 0. IF NOT, SEEK REVERSE 1 CYLINDER AT A TIME UNTIL CYLINDER 0 IS REACHED. IF ANY REVERSE SEEK FAILS TO MOVE THE HEADS IN 10 TRIES:

DETECTION OF GUARD BAND PREMATURE.

WHEN AT CYLINDER 0, DO SEEK DIFFERENCE OF 1, SIGN 0, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR READY. READY SHOULD SET IN 20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR CYLINDER 0. IF NOT

FAILED TO SEEK BACK TO ZERO

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. DO SAME TESTS AS ABOVE WITH REGARD TO READY VS TIME AND CYLINDER FOUND IN

HEADER.

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO THAT SURFACE.

TEST 4 INCREMENTAL FORWARD SEEK HEAD 0 TEST

POSITION HEADS AT CYLINDER "LOLIMIT" USING SEEKS WITH DIFFERENCE OF ONE, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY IS SET IN 15 MS. IF NOT:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
MECHANICAL OBSTRUCTION

DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER + 1 IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEKS AND READS UNTIL CYLINDER READ IS "HILIMIT".

NOTE 1: IF THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 1 IS CHOSEN.

TEST 5 INCREMENTAL REVERSE SEEK HEAD 0 TEST

POSITION HEADS AT CYLINDER "HILIMIT" USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER - 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEK AND CHECKS UNTIL CYLINDER IS "LOLIMIT".

NOTE: IF THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 1 IS CHOSEN.

TEST 6 INCREMENTAL FORWARD SEEK HEAD 1 TEST

POSITION HEADS AT CYLINDER "HILIMIT" USING SEEKS WITH DIFFERENCE OF ONE, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY IS SET IN 15 MS. IF NOT:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER

DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER + 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEKS AND READS UNTIL CYLINDER READ IS "HILIMIT".

NOTE 1: IF THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 0 IS CHOSEN.

TEST 7 INNER GUARD BAND DETECTION TEST

POSITION HEADS AT CYLINDER 255 USING SEEK WITH DIFFERENCE OF 1, HEAD 0.

WHEN AT CYLINDER 255, DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. READY SHOULD SET IN 20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR CYLINDER 255. IF NOT:

FAILED TO SEEK BACK TO CYLINDER 255

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1. DO SAME TESTS AS ABOVE.

N 2

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO THAT SURFACE.

SEQ 0026

TEST 8 INCREMENTAL REVERSE SEEK HEAD 1 TEST

POSITION HEADS AT CYLINDER "HILIMIT" USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER

DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER - 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEK AND CHECKS UNTIL CYLINDER IS "LOLIMIT".

NOTE 1: IF PROGRAM MODE 2 IS USED AND THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 0 IS CHOSEN.

TEST 9 SEEK TESTS

POSITION HEADS AT CYLINDER "LOLIMIT" USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 1, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER + DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE
DIFFERENCE COUNTER FAILURE
COUNT PULSE GENERATION FAILURE
VELOCITY ROM FAILURE

REPEAT ABOVE UNTIL OLD CYLINDER + DISTANCE > 255. POSITION AT 255.

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 0, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER - DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE

REPEAT UNTIL OLD CYLINDER - DISTANCE < 0. REPEAT ALL OF THE ABOVE USING HEAD 1.

REPEAT ALL OF THE ABOVE TESTS USING THE FOLLOWING DISTANCES: 6, 9, 12, 17, 22, 27, 34, 41, 128, 256. THESE DISTANCES ARE SPECIFIED BECAUSE THEY REPRESENT THE MAXIMUM DISTANCE FOR EACH

VELOCITY LEVEL USED IN THE DRIVE.

NOTE: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 10 FORWARD OSCILLATING SEEK TEST

POSITION HEADS AT CYLINDER 0.

DO OSCILLATING SEEK USING HEAD 0 (SEEK FROM 0 TO 1 TO 0, 0 TO 2 TO 0, 0 TO 3 TO 0, ...0 TO 255 TO 0). AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

TEST 11 REVERSE OSCILLATING SEEK TEST

POSITION HEADS AT CYLINDER 255. DO OSCILLATING SEEK USING HEAD 0. (SEEK FROM 255 TO 254 TO 255,255 TO 253 TO 255,....255 TO 0 TO 255.) AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

TEST 12 SEEK TIMING

POSITION HEADS AT CYLINDER 0.

DO 64 SEEKS FROM 0 TO 1 AND 1 TO 0, MEASURING THE SEEK TIME FOR EACH SEEK. AVERAGE THE SEEK TIMES (FORWARD AND REVERSE INDEPENDENTLY) AND REPORT.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 127 TO 128 AND 254 TO 255.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 TO 127 AND 128 TO 256.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 AND 255.

THE SEEK TIMES WILL BE REPORTED AS SHOWN BELOW. THE TIME MEASURED IS FROM START OF SEEK COMMAND UNTIL INTERRUPT IS RECEIVED.

	INNER	MIDDLE	OUTER	EXPECTED
1 CYL FWD	X	X	X	X
1 CYL REV	X	X	X	X

128 CYL FWD	X		X	X
128 CYL REV	X		X	X
256 CYL FWD		X		X
256 CYL REV		X		X

THE X INDICATES WHERE TIME WILL BE REPORTED.

NOTE: THE ABOVE REPORT WILL BE PRINTED IN THE FIRST PASS FOR EACH DRIVE UNDER TEST IF MANUAL INTERVENTION TESTS WERE RUN. THE EXPECTED TIMES ARE FOR USER COMPARISON ONLY. THE PROGRAM WILL NOT REPORT DEVIATION AS AN ERROR.

TEST 13 BASIC READ DATA TEST

POSITION HEADS AT CYLINDER 255.

DO READ DATA, HEAD 1. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 1 THROUGH 19 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ, SUCCESSFULLY, REPORT THAT FACTORY BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READ OF SECTOR 20.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA FORMAT (WORD 0 AND 1 ARE NOT 0, WORD 2 AND 3 ARE 0, LOCATE FIRST WORD OF ALL ONE'S AND THAT WORD TO WORD 127 ARE ALL ONE'S.) STORE BAD SECTOR DATA.

READ DATA, HEAD ONE, SECTOR 20. CHECK FOR ANY ERRORS AND

REPORT. IF ERROR, READ SECTOR 21 THROUGH 39 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ SUCCESSFULLY, REPORT THAT SOFTWARE BAD SECTOR FILES CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT TEST.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA AS ABOVE. STORE BAD SECTOR DATA.

NOTE: IF SURFACE 0 IS SELECTED THIS TEST WILL BE BYPASSED.

TEST 14 WRITE/READ DATA TEST (PART 1)

POSITION HEADS AT CYLINDER 0

WRITE PATTERN 1 ON HEAD 0, SECTOR 0. CHECK FOR ANY ERROR.

READ HEAD 0, SECTOR 0. CHECK FOR CRC ERROR. COMPARE DATA.

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

CHECK IF CYLINDER 0, TRACK 1, SECTOR 0 IS LISTED IN BAD SECTOR DATA. IF NOT, REPEAT ABOVE TEST AT CYLINDER 0, TRACK 1, SECTOR 0. IF IT IS LISTED AS BAD, LOCATE FIRST SECTOR 0, TRACK 1 THAT IS GOOD AND DO ABOVE TESTS.

NOTE: CYLINDER LIMITS ARE IGNORED, TESTING IS DONE AT CYLINDER 0. HOWEVER, CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 15 SPINDLE TIMING TEST

POSITION HEADS TO CYLINDER 0.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. WAIT FOR INTERRUPT.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. START TIMING. WHEN INTERRUPT OCCURS, STOP TIMING. RESULT IS SPINDLE ROTATION TIME.

REPEAT TEST 64 TIMES. REPORT THE AVERAGE AS SPINDLE ROTATION TIME. THE TIME REPORTED IS IN 100'S OR MICROSECONDS.

NOTE: THIS TEST WILL BE RUN ONLY IN THE FIRST PASS AND ONLY IF MANUAL INTERVENTION TESTS WERE RUN.

TEST 16 WRITE/READ TEST (PART 2)

CC IS CURRENT CYLINDER SELECTED FROM SET.
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.

SEEK FORWARD TO CC. WRITE PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ/COMPARE ALL DATA.

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE ALL DATA. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. REWRITE DATA PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ COMPARE ALL DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

NOTE 1: IF ANY OF THE SECTORS IN THE SELECTED CYLINDER SET ARE LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 6 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE EVERY 8TH ENTRY IN THE TABLE. ON THE SECOND AND SUBSEQUENT PASSES ALL ENTRIES IN THE SELECTED CYLINDER SET ARE USED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 17 WRITE LOCK ERROR AND DATA PROTECTION TEST

DO WRITE DATA PATTERN 0 AT SECTOR 0. READ DATA AND VERIFY.

ASK OPERATOR TO WRITE LOCK DRIVE. DO GET STATUS LOOP UNTIL WRITE LOCK IS SET. IF NOT SET IN 30 SECONDS, ABORT THE TEST.

WHEN WRITE LOCK IS SET, DO WRITE DATA PATTERN 1 AT SECTOR 0. REPORT FAILURE IF DRIVE ERROR DOES NOT SET OR IF ANY OTHER ERROR SETS. CLEAR ERROR AND READ DATA AT SECTOR 0. CHECK THAT DATA HAS NOT BEEN DISTURBED.

REQUEST OPERATOR TO RESET WRITE LOCK. DO GET STATUS LOOP UNTIL WRITE LOCK IS RESET. IF NOT RESET IN 30 SECONDS, REPEAT THE REQUEST.

NOTE: THIS TEST IS EXECUTED ONLY IF THE PROGRAM OPERATION MODE 2 IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED, AND IS RUN IN FIRST PASS ONLY.

TEST 18 ADJACENT CYLINDER INTERFERENCE TEST

CC IS CURRENT CYLINDER SELECTED FROM SET
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.
DATA PATTERN IS 155555.

SEEK FORWARD TO CYLINDER CC. WRITE PATTERN ON TRACK 0, ALL SECTORS. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC-1. WRITE PATTERN. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE PATTERN. (THIS HAS BRACKETED ORIGINAL WRITE WITH WRITES IN ADJACENT CYLINDERS. NOTE ADJACENT CYLINDERS WERE WRITTEN AFTER HEADS CAME ON CYLINDER IN REVERSE DIRECTION WHICH IS OPPOSITE OF CENTER CYLINDER.)

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE DATA FROM ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE DATA PATTERN. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC-1, WRITE PATTERN. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC+1, WRITE PATTERN. SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. READ/COMPARE DATA IN ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS (EXCEPT 0 AND 255) IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST, TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON

SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 19 OVERWRITE TEST

CC IS CURRENT CYLINDER SELECTED FROM SET
SELECTED CYLINDER SET DEFINED IN PARAGRAPH 4.3.
PATTERN A = 125252
PATTERN B = 000000

SEEK FORWARD TO CC. WRITE DATA OF PATTERN A IN ALL SECTORS, HEAD 0. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE PATTERN B. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC, READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE DATA PATTERN A. READ/COMPARE DATA. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC. WRITE PATTERN B. SEEK FORWARD TO "HILIMIT" SEEK REVERSE TO CC. READ/COMPARE DATA

ANY FAILURES (READ OR COMPARE) ARE ATTRIBUTED TO OVERWRITE PROBLEM.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST, TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

7.0 PROGRAM LISTING

```
                                                .NLIST  CND,MD,ME
                                                .ENABL  ABS,AMA
                    002000                      .=2000

002000                                          SVC
                    000001                      SVCTST=1
                    000001                      SVCSUB=1
                    000001                      SVCBGL=1
                    000000                      SVCINS=0
                    000000                      SVCTAG=0
002000                                          POINTER BGNSW,BGNSFT,BGNDU

002000                                  BGNMOD  MDHEDR
002000                                  HEADER  DZRLD,A,0
(5) 002000      104                             .ASCII  @D@
(5) 002001      132                             .ASCII  @Z@
(5) 002002      122                             .ASCII  @R@
(5) 002003      114                             .ASCII  @L@
(5) 002004      104                             .ASCII  @D@
(6) 002005      000                             .BYTE   0
(6) 002006      000                             .BYTE   0
(5) 002007      000                             .BYTE   0
(4) 002010      101                             .ASCII  @A@
(4) 002011      060                             .ASCII  @0@
(4) 002012      001                             .BYTE   C$REVISION
(3) 002013      006                             .BYTE   C$EDIT
(4) 002014  000000                              .WORD   0
(4) 002016  000000                              .WORD
(4) 002020  000000                              .WORD
(4) 002022  000000                              .WORD
(4) 002024  000000                              .WORD   0
(5) 002026  000000                              .WORD   0
(4) 002030  000000                              .WORD   0
(4) 002032  000000                              .WORD   0
(4) 002034  000000                              .WORD   0
(4) 002036  000000                              .WORD   0
(4) 002040  013200                              .WORD   L$DISPATCH
(4) 002042  013246                              .WORD   L$INIT
(4) 002044  014200                              .WORD   L$CLEAN
(4) 002046  037340                              .WORD   L$HARD
(4) 002050  037464                              .WORD   L$SOFT
(4) 002052  002104                              .WORD   L$DVTYP
(4) 002054  000000                              .WORD   0
(4) 002056  013146                              .WORD   L$HW
(4) 002060  013162                              .WORD   L$SW
(4) 002062  002102                              .WORD   L$DR
(4) 002064  002102                              .WORD   L$DRST
(4) 002066  000000                              .WORD   0
(4) 002070  000000                              .WORD   0
(4) 002072  014320                              .WORD   L$DU
(4) 002074  000000                              .WORD   0
(4) 002076  040176                              .WORD   L$LAST
2826 002100                             ENDMOD
2827 002100                                     DEVREG
(5) 002100  000000                              .WORD   0
(2) 002102  000001                              .BLKW
```

```
 2828  002104                                    DEVTYP  <RL01>
  (3)  002104  046122  030460     000           .ASCIZ  @RL01@
  (2)          002112                           .EVEN
 2829
 2830                                            ;COPYRIGHT (C) 1977
 2831                                            ;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
 2832                                            ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
 2833                                            ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
 2834                                            ;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
 2835                                            ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
 2836                                            ;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
 2837                                            ;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
 2838                                            ;AT ALL TIMES REMAIN IN DEC.
 2839                                            ;
 2840                                            ;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
 2841                                            ;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
 2842                                            ;BY DIGITAL EQUIPMENT CORPORATION.
 2843                                            ;
 2844                                            ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
 2845                                            ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
 2846
 2847  002112                            BGNMOD  GLBEQAT
 2848
 2849  002112                            EQUALS
  (1)                                    ;
  (1)                                    ; BIT DIFINITIONS
  (1)                                    ;
  (1)          100000                    BIT15== 100000
  (1)          040000                    BIT14== 40000
  (1)          020000                    BIT13== 20000
  (1)          010000                    BIT12== 10000
  (1)          004000                    BIT11== 4000
  (1)          002000                    BIT10== 2000
  (1)          001000                    BIT09== 1000
  (1)          000400                    BIT08== 400
  (1)          000200                    BIT07== 200
  (1)          000100                    BIT06== 100
  (1)          000040                    BIT05== 40
  (1)          000020                    BIT04== 20
  (1)          000010                    BIT03== 10
  (1)          000004                    BIT02== 4
  (1)          000002                    BIT01== 2
  (1)          000001                    BIT00== 1
  (1)                                    ;
  (1)          001000                    BIT9==  BIT09
  (1)          000400                    BIT8==  BIT08
  (1)          000200                    BIT7==  BIT07
  (1)          000100                    BIT6==  BIT06
  (1)          000040                    BIT5==  BIT05
  (1)          000020                    BIT4==  BIT04
  (1)          000010                    BIT3==  BIT03
  (1)          000004                    BIT2==  BIT02
  (1)          000002                    BIT1==  BIT01
  (1)          000001                    BIT0==  BIT00
  (1)                                    ;
  (1)                                    ; EVENT FLAG DEFINITIONS
```

```
(1)                             ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1)                             ;   EF16:EF01 AVAILABLE FOR PROGRAM USE
(1)                             ;
(1)          000040             EF.START==      32.                     ; START COMMAND WAS ISSUED
(1)          000037             EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
(1)          000036             EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
(1)          000035             EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
(1)          000034             EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
(1)                             ;
(1)          000020             EF16==  16.
(1)          000017             EF15==  15.
(1)          000016             EF14==  14.
(1)          000015             EF13==  13.
(1)          000014             EF12==  12.
(1)          000013             EF11==  11.
(1)          000012             EF10==  10.
(1)          000011             EF09==  9.
(1)          000010             EF08==  8.
(1)          000007             EF07==  7
(1)          000006             EF06==  6
(1)          000005             EF05==  5
(1)          000004             EF04==  4
(1)          000003             EF03==  3
(1)          000002             EF02==  2
(1)          000001             EF01==  1
(1)                             ;
(1)
(1)                             ; PRIORITY LEVEL DEFINITIONS
(1)                             ;
(1)          000340             PRI07== 340
(1)          000300             PRI06== 300
(1)          000240             PRI05== 240
(1)          000200             PRI04== 200
(1)          000140             PRI03== 140
(1)          000100             PRI02== 100
(1)          000040             PRI01== 40
(1)          000000             PRI00== 0
2850                            ;       OFFSETS FOR HARDWARE P-TABLE
2851         000000             CSR     =0                      ;BUS ADDRESS
2852         000002             VECT    =2                      ;VECTOR ADDRESS
2853         000004             PRIOR   =4                      ;PRIORITY
2854         000006             DRSB    =6                      ;DRIVE SELECT BIT
2855         000010             CNT     =10                     ;CONTROLLER TYPE
2856
2857                            ;       OFFSET FOR SOFTWARE P-TABLE
2858         000000             MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
2859         000002             LOLIM   =2                      ;CYLINDER LOWER LIMIT
2860         000004             HILIM   =4                      ;CYLINDER HIGH LIMIT
2861         000006             HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
2862         000010             ERLIM   =10                     ;ERROR LIMIT
2863         000012             DCLIM   =12                     ;DATA COMPARE ERROR LIMIT
2864
2865                            ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
2866         000001             ALLCYL  =BIT00                  ;USE ALL CYLINDERS
2867         000002             ALLSEC  =BIT01                  ;USE ALL SECTORS
2868         000004             DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
```

```
                                HDALIGN =BIT03          ;EXECUTE HEAD ALIGNMENT TEST
                                AUTOSZ  =BIT04          ;AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
                                HEADLM  =BIT12          ;HEAD LIMIT SPECIFIED FLAG
                                HICYL   =BIT13          ;HI LIMIT SPECIFIED FLAG
                                LOCYL   =BIT14          ;LO LIMIT SPECIFIED
                                MITEST  =BIT15          ;EXECUTE MANUAL INTERVENTION TESTS

                                ;       SUBSYSTEM FUNCTIONS
                                CKDATA  =102            ;WRITE CHECK
                                GTSTAT  =104            ;GET STATUS
                                SEEK    =106            ;SEEK
                                RDHEAD  =110            ;READ HEADER
                                WTDATA  =112            ;WRITE DATA
                                RDDATA  =114            ;READ DATA
                                RDNOHR  =116            ;READ DATA, IGNORE HEADERS
                                NOOP    =100            ;NO OPERATION

                                ;       OPERATION FLAGS
                                COMPOP  =7777           ;COMPOSITE OPERATION FLAGS
                                HDRCMP  =BIT01          ;HEADER COMPARE OPERATION
                                DATACMP =BIT00          ;DATA COMPARE OPERATION
                                CYLUP   =BIT02          ;CYCLE UP OPERATION
                                ULOAD   =BIT03          ;UNLOAD OPERATION
                                INOUTS  =BIT04          ;IN-OUT SEEK OPERATION
                                OUTINS  =BIT05          ;OUT-IN SEEK OPERATION
                                FOLWRT  =BIT06          ;FOLLOWING WRITE OPERATION
                                REVSKS  =BIT07          ;REV SEEK SEQ (ADJ INTERFERENCE)
                                FWDSKS  =BIT08          ;FWD SEEK SEQ (ADJ INTERFERENCE)
                                REVSKO  =BIT09          ;REV SEEK SEQ (OVERWRITE)
                                FWDSKO  =BIT10          ;FWD SEEK SEQ (OVERWRITE)
                                BADADD  =BIT11          ;BAD DISK ADDRESS
                                SEEKOP  =BIT12          ;SEEK OPERATION
                                RORWOP  =BIT13          ;READ OR WRITE OPERATION
                                RELDWT  =BIT14          ;RELOAD WAIT
                                HDR40   =BIT15          ;40 HEADER OPERATION
                                MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO
                                                        ;MESSAGE QUALIFIER BITS

                                ;       ERROR FLAGS FROM SUBROUTINES
                                TOSLOW  =BIT00          ;OPERATION TOOK TO LONG
                                NOIRPT  =BIT01          ;NO INTERRUPT FROM OPERATION
                                CONHNG  =BIT02          ;CONTROLLER HUNG
                                NOCLR   =BIT03          ;BAD CONTROLLER CLEAR

                                RLCS    =0              ;CONTROL AND STATUS REGISTER
                                RLBA    =2              ;BUS ADDRESS REGISTER
                                RLDA    =4              ;DISK ADDRESS REGISTER
                                RLMP    =6              ;MULTI-PURPOSE REGISTER

                                ;       REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
                                RLCSR   =0              ;CONTROL AND STATUS REGISTER
                                ANYERR  =100000         ;ANY ERROR BIT
                                DRVERR  =40000          ;DRIVE ERROR BIT
                                NXMERR  =20000          ;NON-EXISTANT MEMORY ERROR
                                DLTERR  =10000          ;DATA LATE ERROR
                                HNFERR  =10000          ;HEADER NOT FOUND ERROR
```

```
                                DCKERR  =4000                  ;DATA CHECK ERROR
                                HCRCERR =4000                  ;HEADER CHECK ERROR
                                OPIERR  =2000                  ;OPERATION INCOMPLETE ERROR
                                DSMSK   =1400                  ;DRIVE SELECT MASK
                                CRDYMSK =200                   ;CONTROLLER READY MASK
                                INTEBL  =100                   ;INTERRUPT ENABLE MASK
                                BAMSK   =60                    ;BUS ADDRESS UPPER MASK
                                DRDYMSK =1                     ;DRIVE READY MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
                                SAMSK   =77                    ;SECTOR ADDRESS MASK
                                HSMSK   =100                   ;HEAD SELECT MASK
                                CAMSK   =77600                 ;CYLINDER ADDRESS MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
                                MBSET0  =1                     ;MUST BE SET, BIT 0
                                DIRBIT  =4                     ;DIRECTION BIT
                                HDSEL   =20                    ;HEAD SELECT BIT
                                DIRMSK  =77600                 ;CYLINDER DIFFERENCE MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
                                GETSTAT =3                     ;GET STATUS SETUP
                                DRSET   =10                    ;DRIVE RESET MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR DATA XFER
                                WCMSK   =17777                 ;WORD COUNT MASK
                                WCRNG   =160000                ;WORD COUNT RANGE MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR READ HEADER
                                HDCYL   =077600                ;CYLINDER MASK
                                HDSEC   =77                    ;SECTOR MASK
                                HDHSEL  =100                   ;HEAD SELECT MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
                                STAMSK  =7                     ;STATE MASK
                                BHSTAT  =10                    ;BRUSH HOME STATUS
                                HOSTAT  =20                    ;HEADS OUT STATUS
                                COSTAT  =40                    ;COVER OPEN STATUS
                                HSSTAT  =100                   ;HEAD SELECT STATUS
                                DSESTAT =400                   ;DRIVE SELECT ERROR STATUS
                                VCSTAT  =1000                  ;VOLUME CHECK STATUS
                                WGESTAT =2000                  ;WRITE GATE ERROR STATUS
                                SPDSTAT =4000                  ;SPIN ERROR STATUS
                                STOSTAT =10000                 ;SEEK TIMEOUT ERROR STATUS
                                WLSTAT  =20000                 ;WRITE LOCK STATUS
                                HCESTAT =40000                 ;HEAD CURRENT ERROR STATUS
                                WDESTAT =100000                ;WRITE DATA ERROR STATUS

002112                          ENDMOD
002112                          BGNMOD  GLBDAT

                                ;       TABLE OF OPERATION MESSAGES
002112  000000                  OPMSGS: .WORD   0              ;FILLER
002114  004572                          .WORD   MWRCHK         ;MESSAGE FOR WRITE CHECK
002116  004622                          .WORD   MGTSTA         ;             GET STATUS
002120  004534                          .WORD   MSEEK          ;             SEEK
```

```
2981  002122  004555                          .WORD   MREADH         ;           READ HEADER
2982  002124  004606                          .WORD   MWRITE         ;           WRITE DATA
2983  002126  004542                          .WORD   MREAD          ;           READ DATA
2984  002130  004717                          .WORD   MWRSET         ;           WITH RESET
2985  002132  004636                          .WORD   MDATCP         ;           WITH DATA COMPARE
2986  002134  004661                          .WORD   MHDRCP         ;           WITH HEADER COMPARE
2987  002136  004774                          .WORD   MCYLUP         ;           LOAD HEADS
2988  002140  004763                          .WORD   MULOAD         ;           UNLOAD HEADS
2989  002142  005025                          .WORD   MINOUT         ;              IN-OUT SEQ
2990  002144  005004                          .WORD   MOUTIN         ;              OUT-IN SEQ
2991  002146  005050                          .WORD   MFOLWRT        ;              FOLLOWING WRITE
2992  002150  005074                          .WORD   MREVSK         ;              REV SEEK
2993  002152  005127                          .WORD   MFWDSK         ;              FWD SEEK
2994  002154  005216                          .WORD   MRESKO         ;              REV SEEK
2995  002156  005162                          .WORD   MFWSKO         ;              FWD SEEK
2996  002160  005252                          .WORD   MBADAD         ;              BAD DISK ADD FOR WRITE
2997  002162  004703                          .WORD   M40HDR         ;           40 HEADER OPERATION
2998
2999                                  ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
3000  002164  007344                  RESTBL: .WORD   MCERR           ;CONTROLLER ERROR
3001  002166  007455                          .WORD   MDRERR         ;DRIVE ERROR
3002  002170  007704                          .WORD   MNEERR         ;NON-EXISTANT MEMORY ERROR
3003  002172  007656                          .WORD   MFLERR         ;HEADER NOT FOUND-DATA LATE
3004  002174  007641                          .WORD   MHDERR         ;HEADER OR DATA ERROR
3005  002176  007622                          .WORD   MOPERR         ;OPERATION INCOMPLETE
3006  002200  007731                          .WORD   MNDRST         ;NO DRIVE STATUS AVAILABLE
3007  002202  000000                          .WORD   0
3008  002204  007605                          .WORD   MWDERR         ;WRITE DATA ERROR
3009  002206  007567                          .WORD   MHCERR         ;HEAD CURRENT ERROR
3010  002210  000000                          .WORD   0
3011  002212  007551                          .WORD   MSTERR         ;SEEK TIMEOUT ERROR
3012  002214  007516                          .WORD   MSPERR         ;SPINDLE ERROR
3013  002216  007534                          .WORD   MWGERR         ;WRITE GATE ERROR
3014  002220  000000                          .WORD   0
3015  002222  007466                          .WORD   MDSERR         ;DRIVE SELECT ERROR
3016
3017                                  ;       PATTERN TABLE
3018  002224  004256                  PATTBL: .WORD   PAT1
3019  002226  004260                          .WORD   PAT2
3020  002230  004320                          .WORD   PAT3
3021  002232  004360                          .WORD   PAT4
3022  002234  004420                          .WORD   PAT5
3023  002236  004426                          .WORD   PAT6
3024  002240  004466                          .WORD   PAT7
3025  002242  004470                          .WORD   PAT8
3026  002244  004530                          .WORD   PAT9
3027  002246  004532                          .WORD   PAT10
3028
3029
3030                                  ;       SUBROUTINE CALLING STACK
3031  002250  000000                  SUBSTK: .WORD   0               ;STACK IS 12 WORDS LONG
3032  002252  000000                          .WORD   0
3033  002254  000000                          .WORD   0
3034  002256  000000                          .WORD   0
3035  002260  000000                          .WORD   0
3036  002262  000000                          .WORD   0
```

```
3037  002264  000000                          .WORD   0
3038  002266  000000                          .WORD   0
3039  002270  000000                          .WORD   0
3040  002272  000000                          .WORD   0
3041
3042  002274  000002                T25TBL:   .WORD   2               ;TABLE OF DIFFERENCES TO BE USED
3043  002276  000006                          .WORD   6               ;IN TEST 25
3044  002300  000011                          .WORD   9.
3045  002302  000014                          .WORD   12.
3046  002304  000021                          .WORD   17.
3047  002306  000026                          .WORD   22.
3048  002310  000033                          .WORD   27.
3049  002312  000042                          .WORD   34.
3050  002314  000051                          .WORD   41.
3051  002316  000200                          .WORD   128.
3052  002320  000377                          .WORD   255.
3053
3054                                  ;       TABLE TO BE USED IN TEST 33 AND 34 TO BUILD AND STORE THE
3055                                  ;       CYLINDERS TO BE USED IN THE TEST.
3056  002322  000010                T33TBL:   .BLKW   10
3057
3058  002342      002                CYLTBL:   .BYTE   2               ;TABLE OF DEFAULT CYLINDERS
3059  002343      007                          .BYTE   7.
3060  002344      016                          .BYTE   14.
3061  002345      024                          .BYTE   20.
3062  002346      033                          .BYTE   27.
3063  002347      041                          .BYTE   33.
3064  002350      046                          .BYTE   38.
3065  002351      055                          .BYTE   45.
3066  002352      064                          .BYTE   52.
3067  002353      072                          .BYTE   58.
3068  002354      101                          .BYTE   65.
3069  002355      110                          .BYTE   72.
3070  002356      115                          .BYTE   77.
3071  002357      124                          .BYTE   84.
3072  002360      133                          .BYTE   91.
3073  002361      141                          .BYTE   97.
3074  002362      146                          .BYTE   102.
3075  002363      154                          .BYTE   108.
3076  002364      161                          .BYTE   113.
3077  002365      170                          .BYTE   120.
3078  002366      177                          .BYTE   127.
3079  002367      206                          .BYTE   134.
3080  002370      213                          .BYTE   139.
3081  002371      222                          .BYTE   146.
3082  002372      230                          .BYTE   152.
3083  002373      235                          .BYTE   157.
3084  002374      244                          .BYTE   164.
3085  002375      252                          .BYTE   170.
3086  002376      261                          .BYTE   177.
3087  002377      270                          .BYTE   184.
3088  002400      275                          .BYTE   189.
3089  002401      303                          .BYTE   195.
3090  002402      312                          .BYTE   202.
3091  002403      317                          .BYTE   207.
3092  002404      326                          .BYTE   214.
```

```
3093  002405  334                         .BYTE   220.
3094  002406  343                         .BYTE   227.
3095  002407  352                         .BYTE   234.
3096  002410  361                         .BYTE   241.
3097  002411  367                         .BYTE   247.
3098  002412  375                         .BYTE   253.
3099  002413  000                         .BYTE   0
3100
3101  002414  000000              SSINDX: .WORD   0               ;SUBROUTINE STACK INDEX POINTER
3102
3103                              ;       OPERATIONAL FLAGS
3104  002416  000000              OPFLAG: .WORD   0               ;OPERATION FLAGS
3105  002420  000000              DONE:   .WORD   0               ;OPERATION COMPLETE FLAG
3106  002422  000000              HADONE: .WORD   0               ;HEAD ALIGNMENT DONE FLAG
3107  002424  000000              ERHEAD: .WORD   0               ;ADDRESS OF ERROR HEADER
3108  002426  000000              MORECE: .WORD   0               ;MORE THAN 1 COMPARE ERROR
3109  002430  000000              ERRSWI: .WORD   0               ;ERROR RETURN SWITCH
3110  002432  000000              BSFLAG: .WORD   0               ;BAD SECTOR FLAGS
3111  002434  000000              WRTSWI: .WORD   0               ;WRITE SWITCH
3112  002436  000000              TBLSTR: .WORD   0               ;TABLE STORAGE
3113
3114  002440  000000              RLBAS:  .WORD   0               ;RL11 BASE ADDRESS
3115  002442  000000              RLVEC:  .WORD   0               ;RL11 VECTOR ADDRESS
3116  002444  000000              RLDRV:  .WORD   0               ;DRIVE NUMBER UNDER TEST
3117
3118  002446  000000              L.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
3119  002450  000000              L.BA:   .WORD   0               ;BEFORE OPERATION
3120  002452  000000              L.DA:   .WORD   0
3121  002454  000000              L.MP:   .WORD   0
3122  002456  000000              T.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
3123  002460  000000              T.BA:   .WORD   0               ;   AFTER OPERATION
3124  002462  000000              T.DA:   .WORD   0
3125  002464                      T.MP:
3126  002464  000000              HDWRD1: .WORD   0               ;HEADER WORD STORAGE
3127  002466  000000              HDWRD2: .WORD   0
3128  002470  000000              HDWRD3: .WORD   0
3129
3130  002472  000000              T.STAT: .WORD   0               ;DRIVE STATE STORAGE
3131
3132  002474  000000              RESPARM:        .WORD   0       ;PARAM BLOCK FOR REASON REPORT
3133  002476  000000                      .WORD   0
3134  002500  000000                      .WORD   0
3135  002502  000000                      .WORD   0
3136  002504  000000                      .WORD   0
3137
3138  002506  000000              DRVCNT: .WORD   0               ;DRIVE COUNT FOR DRIVES UNDER TEST
3139  002510  000000              DIFAUG: .WORD   0               ;DIFFERENCE AUGMENT FOR SEEK
3140  002512  000000              OLDCYL: .WORD   0               ;OLD CYLINDER
3141  002514  000000              NEWCYL: .WORD   0               ;NEW CYLINDER
3142  002516  000000              CURCYL: .WORD   0               ;CURRENT CYLINDER
3143  002520  000000              DESDIF: .WORD   0               ;DESIRED DIFFERENCE
3144  002522  000000              DESSGN: .WORD   0               ;DESIRED SIGN
3145  002524  000000              DESHD:  .WORD   0               ;DESIRED HEAD
3146  002526  000000              DESSEC: .WORD   0               ;DESIRED SECTOR
3147  002530  000000              TEMP0:  .WORD   0               ;TEMPORARY STORAGE
3148  002532  000000              TEMP1:  .WORD   0               ;TEMPORARY STARAGE
```

```
3149  002534  000000                TEMP2:  .WORD   0               ;TEMPORARY STORAGE
3150  002536  000000                TEMP3:  .WORD   0               ;TEMPORARY STORAGE
3151  002540  000000                TEMP4:  .WORD   0               ;TEMPORARY STORAGE
3152  002542  000000                TEMP5:  .WORD   0               ;TEMPORARY STORAGE
3153  002544  000000                TEMP6:  .WORD   0               ;TEMPORARY STORAGE
3154  002546  000000                TEMP7:  .WORD   0               ;TEMPORARY STORAGE
3155  002550  000000                TEMP8:  .WORD   0               ;TEMPORARY STORAGE
3156
3157                                ;       TIMER STORAGE
3158  002552  000000                OFIN:   .WORD   0               ;ONE CYLINDER FORWARD INNER
3159  002554  000000                OFINU:  .WORD   0               ;       UPPER
3160  002556  000000                OFMID:  .WORD   0               ;ONE CYLINDER FORWARD MIDDLE
3161  002560  000000                OFMIDU: .WORD   0               ;       UPPER
3162  002562  000000                OFOUT:  .WORD   0               ;ONE CYLINDER FORWARD OUTER
3163  002564  000000                OFOUTU: .WORD   0               ;       UPPER
3164  002566  000000                ORIN:   .WORD   0               ;ONE CYLINDER REVERSE INNER
3165  002570  000000                ORINU:  .WORD   0               ;       UPPER
3166  002572  000000                ORMID:  .WORD   0               ;ONE CYLINDER REVERSE MIDDLE
3167  002574  000000                ORMIDU: .WORD   0               ;       UPPER
3168  002576  000000                OROUT:  .WORD   0               ;ONE CYLINDER REVERSE OUTER
3169  002600  000000                OROUTU: .WORD   0               ;       UPPER
3170  002602  000000                HFIN:   .WORD   0               ;128 CYLINDER FORWARD INNER
3171  002604  000000                HFINU:  .WORD   0               ;       UPPER
3172  002606  000000                HFOUT:  .WORD   0               ;128 CYLINDER FORWARD OUTER
3173  002610  000000                HFOUTU: .WORD   0               ;       UPPER
3174  002612  000000                HRIN:   .WORD   0               ;128 CYLINDER REVERSE INNER
3175  002614  000000                HRINU:  .WORD   0               ;       UPPER
3176  002616  000000                HROUT:  .WORD   0               ;128 CYLINDER REVERSE OUTER
3177  002620  000000                HROUTU: .WORD   0               ;       UPPER
3178  002622  000000                AFMID:  .WORD   0               ;256 CYLINDER FORWARD
3179  002624  000000                AFMIDU: .WORD   0               ;       UPPER
3180  002626  000000                ARMID:  .WORD   0               ;256 CYLINDER REVERSE
3181  002630  000000                ARMIDU: .WORD   0               ;       UPPER
3182
3183  002632  000226                EXOCYL: .WORD   150.            ;EXPECTED TIME ONE CYLINDER
3184  002634  001046                EXHCYL: .WORD   550.            ;EXPECTED TIME 128 CYLINDER
3185  002636  001750                EXACYL: .WORD   1000.           ;EXPECTED TIME 256 CYLINDER
3186  002640  000372                EXROT:  .WORD   250.            ;EXPECTED ROTATION TIME
3187  002642  000004                ERRVEC: .WORD   4               ;ERROR VECTOR USED WHEN AUTO SIZING
3188
3189                                ;       MISCELLANEOUS COUNTERS
3190  002644  000000                PASCNT: .WORD   0               ;PASS COUNTER (LOCAL TO A TEST)
3191  002646  000000                COUNT:  .WORD   0               ;A COUNTER (LOCAL TO A TEST)
3192  002650  000000                ERRCNT: .WORD   0               ;ERROR COUNTER FOR PROGRAM
3193  002652  000000                PASNUM: .WORD   0               ;PASS NUMBER FOR PROGRAM
3194  002654  000000                PSETNM: .WORD   0               ;COUNTER FOR PARAMETER SET NUMBER IN USE
3195  002656     000                LOCERR: .BYTE   0               ;LOCAL ERROR COUNTER
3196  002657     000                NOERCT: .BYTE   0               ;INHIBIT ERROR COUNTING FLAG
3197  002660  000000                TRPFLG: .WORD   0               ;HARDWAPE TRAP OCCURANCE
3198  002662  000000                PWRFLG: .WORD   0               ;POWER FAILURE OCCURANCE
3199
3200                                ;       BAD SECTOR TABLES AND POINTERS
3201  002664  000000                BSFVAL: .WORD   0               ;BAD SECTORS FILES VALID FLAG
3202
3203  002666  000076                SBSFIL: .BLKW   76              ;SOFTWARE BAD SECTOR FILE
3204  003062  000076                FBSFIL: .BLKW   76              ;FACTORY BAD SECTOR FILE
```

```
3205
3206  003256  000200                IBUFF:  .BLKW   200             ;INPUT BUFFER
3207  003656  000200                OBUFF:  .BLKW   200             ;OUTPUT BUFFER
3208
3209  004256  000000                PAT1:   .WORD   0               ;PATTERN 1 (ALL ZEROS)
3210  004260  177772                PAT2:   .WORD   177772
3211  004262  177777                        .WORD   177777
3212  004264  177777                        .WORD   177777
3213  004266  052525                        .WORD   052525
3214  004270  052525                        .WORD   052525
3215  004272  052525                        .WORD   052525
3216  004274  177777                        .WORD   177777
3217  004276  177777                        .WORD   177777
3218  004300  052525                        .WORD   052525
3219  004302  052525                        .WORD   052525
3220  004304  177777                        .WORD   177777
3221  004306  052525                        .WORD   052525
3222  004310  177252                        .WORD   177252
3223  004312  177252                        .WORD   177252
3224  004314  172765                        .WORD   172765
3225  004316  172765                        .WORD   172765
3226
3227  004320  000003                PAT3:   .WORD   000003
3228  004322  000000                        .WORD   000000
3229  004324  000000                        .WORD   000000
3230  004326  177777                        .WORD   177777
3231  004330  177777                        .WORD   177777
3232  004332  177777                        .WORD   177777
3233  004334  000000                        .WORD   000000
3234  004336  000000                        .WORD   000000
3235  004340  177777                        .WORD   177777
3236  004342  177777                        .WORD   177777
3237  004344  000000                        .WORD   000000
3238  004346  177777                        .WORD   177777
3239  004350  000000                        .WORD   000000
3240  004352  177777                        .WORD   177777
3241  004354  000000                        .WORD   000000
3242  004356  177777                        .WORD   177777
3243
3244  004360  025252                PAT4:   .WORD   025252
3245  004362  052525                        .WORD   052525
3246  004364  052525                        .WORD   052525
3247  004366  125252                        .WORD   125252
3248  004370  125252                        .WORD   125252
3249  004372  125252                        .WORD   125252
3250  004374  052525                        .WORD   052525
3251  004376  052525                        .WORD   052525
3252  004400  125252                        .WORD   125252
3253  004402  125252                        .WORD   125252
3254  004404  052525                        .WORD   052525
3255  004406  125252                        .WORD   125252
3256  004410  052525                        .WORD   052525
3257  004412  125252                        .WORD   125252
3258  004414  052525                        .WORD   052525
3259  004416  125252                        .WORD   125252
3260
```

```
  3261  004420  155555                  PAT5:   .WORD   155555
  3262  004422  133333                          .WORD   133333
  3263  004424  066666                          .WORD   066666
  3264
  3265  004426  121105                  PAT6:   .WORD   121105
  3266  004430  150442                          .WORD   150442
  3267  004432  064221                          .WORD   064221
  3268  004434  132110                          .WORD   132110
  3269  004436  055044                          .WORD   055044
  3270  004440  026442                          .WORD   026442
  3271  004442  013211                          .WORD   013211
  3272  004444  105504                          .WORD   105504
  3273  004446  042642                          .WORD   042642
  3274  004450  021321                          .WORD   021321
  3275  004452  110550                          .WORD   110550
  3276  004454  044264                          .WORD   044264
  3277  004456  022132                          .WORD   022132
  3278  004460  011055                          .WORD   011055
  3279  004462  104426                          .WORD   104426
  3280  004464  042213                          .WORD   042213
  3281
  3282  004466  177777                  PAT7:   .WORD   177777
  3283
  3284  004470  045513                  PAT8:   .WORD   045513
  3285  004472  122645                          .WORD   122645
  3286  004474  151322                          .WORD   151322
  3287  004476  064551                          .WORD   064551
  3288  004500  132264                          .WORD   132264
  3289  004502  055132                          .WORD   055132
  3290  004504  026455                          .WORD   026455
  3291  004506  113226                          .WORD   113226
  3292  004510  045513                          .WORD   045513
  3293  004512  122645                          .WORD   122645
  3294  004514  151322                          .WORD   151322
  3295  004516  064551                          .WORD   064551
  3296  004520  132264                          .WORD   132264
  3297  004522  055132                          .WORD   055132
  3298  004524  026455                          .WORD   026455
  3299  004526  113226                          .WORD   113226
  3300
  3301  004530  125252                  PAT9:   .WORD   125252
  3302
  3303  004532  155555                  PAT10:  .WORD   155555
  3304
  3305  004534                          ENDMOD
  3306
  3310  004534                          BGNMOD  GLBTXT
  3311  004534  042523  045505  000040 MSEEK:  .ASCIZ  /SEEK /
  3312  004542  042522  042101  042040 MREAD:  .ASCIZ  /READ DATA /
  3313  004555     122  040505  020104 MREADH: .ASCIZ  /READ HEADER /
  3314  004572  051127  052111  020105 MWRCHK: .ASCIZ  /WRITE CHECK/
  3315  004606  051127  052111  020105 MWRITE: .ASCIZ  /WRITE DATA /
  3316  004622  042507  020124  052123 MGTSTA: .ASCIZ  /GET STATUS /
  3317  004636  044527  044124  042040 MDATCP: .ASCIZ  /WITH DATA COMPARE /
  3318  004661     127  052111  020110 MHDRCP: .ASCIZ  /WITH HDR COMPARE /
  3319  004703     106  051117  032040 M40HDR: .ASCIZ  /FOR 40 HDRS/
```

```
 3320  004717     127  052111  020110  MWRSET: .ASCIZ  /WITH RESET /
 3321  004733     117  042520  040522  MOPER:  .ASCIZ  /OPERATION: /
 3322  004747     122  051505  046125  MRSLT:  .ASCIZ  /RESULT:    /
 3323  004763     125  046116  020104  MULOAD: .ASCIZ  /UNLD DRV/
 3324  004774  042114  042040  053122  MCYLUP: .ASCIZ  /LD DRV /
 3325  005004  047506  020114  020060  MOUTIN: .ASCIZ  /FOL 0 TO CC SEEK/
 3326  005025     106  046117  031040  MINOUT: .ASCIZ  /FOL 255 TO CC SEEK/
 3327  005050  047506  020114  051127  MFOLWRT:        .ASCIZ  /FOL WRITE (NO SEEK)/
 3328  005074  042101  020112  054503  MREVSK: .ASCIZ  /ADJ CYL WRTTN AFTER REV SK/
 3329  005127     101  045104  041440  MFWDSK: .ASCIZ  /ADJ CYL WRTTN AFTER FWD SK/
 3330  005162  045523  043040  042127  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
 3331  005216  045523  051040  053105  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
 3332  005252  047117  041040  042101  MBADAD: .ASCIZ  /ON BAD SEC FILES/
 3333  005273     103  047101  052047  MBADSF: .ASCIZ  /CAN'T GET BAD SEC FILES/
 3334  005323     102  042101  051440  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
 3335  005350  047524  046440  047101  MTMBS:  .ASCIZ  /TO MANY BAD SEC FOR PROG CAPACITY/
 3336  005412  052502  020123  042101  BASADD: .ASCIZ  /BUS ADD=/
 3337  005423     104  053122  000075  DRVNAM: .ASCIZ  /DRV=/
 3338  005430  051104  053111  020105  DRVNAV: .ASCIZ  /DRIVE UNAVAILABLE FOR TEST/
 3339  005463     104  053122  042040  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
 3340  005523     122  041514  000123  CSNAM:  .ASCIZ  /RLCS/
 3341  005530  046122  040502     000  BANAM:  .ASCIZ  /RLBA/
 3342  005535     122  042114  000101  DANAM:  .ASCIZ  /RLDA/
 3343  005542  046122  050115     000  MPNAM:  .ASCIZ  /RLMP/
 3344  005547     117  020120  047111  LAB1:   .ASCIZ  /OP INIT = /
 3345  005562  050117  042040  047117  LAB2:   .ASCIZ  /OP DONE = /
 3346  005575     127  051117  020104  MWORD:  .ASCIZ  /WORD /
 3347  005603     111  052116  050122  MTOSLOW:        .ASCIZ  /INTRPT TO LATE/
 3348  005622  050117  020111  042523  MDRRES: .ASCIZ  /OPI SET-NO DRV RESPONSE/
 3349  005652  047516  044440  052116  MNOINT. .ASCIZ  /NO INTRPT ON CMND COMPLETE/
 3350  005705     103  052116  051114  MCONHNG:        .ASCIZ  /CNTLR HUNG (NO RDY)/
 3351  005731     105  051122  042040  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
 3352  005751     126  046117  041440  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
 3353  005772  047125  050130  052103  UNXERR: .ASCIZ  /UNXPCTED ERR/
 3354  006007     040  042524  052123  TSTLAB: .ASCIZ  / TEST/
 3369  006015                          P2TO1E:
 3370  006015     104  043111  020106  P2TO2E: .ASCIZ  /DIFF OF 1 SEEK/
 3371  006034  052517  020124  051107  P2TO3E: .ASCIZ  /OUT GRD BAND DETECT/
 3372  006060  047111  020103  042523  P2TO4E: .ASCIZ  /INC SEEK FWD HD 0/
 3373  006102  047111  020103  042523  P2TO5E: .ASCIZ  /INC SEEK REV HD 0/
 3374  006124  047111  020103  042523  P2TO6E: .ASCIZ  /INC SEEK FWD HD 1/
 3375  006146  047111  020116  051107  P2TO7E: .ASCIZ  /INN GRD BAND DETECT/
 3376  006172  047111  020103  042523  P2TO8E: .ASCIZ  /INC SEEK REV HD 1/
 3377  006214  042523  045505     000  P2TO9E: .ASCIZ  /SEEK/
 3378  006221     106  042127  047440  P2T10E: .ASCIZ  /FWD OSC SEEK/
 3379  006236  042522  020126  051517  P2T11E: .ASCIZ  /REV OSC SEEK/
 3380  006253     123  042505  020113  P2T12E: .ASCIZ  /SEEK TIMING/
 3381  006267     102  051501  041511  P2T13E: .ASCIZ  /BASIC READ DATA/
 3382  006307     127  052122  051057  P2T14E: .ASCIZ  &WRT/READ DATA (P1)&
 3383  006332  050123  047111  046104  P2T15E: .ASCIZ  /SPINDLE ROTATION TIMING/
 3384  006362  051127  027524  042522  P2T16E: .ASCIZ  &WRT/READ DATA (P2)&
 3385  006405     127  052122  046040  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROTECTION/
 3386  006445     101  045104  041440  P2T18E: .ASCIZ  /ADJ CYL INTERFERENCE/
 3387  006472  053117  051105  051127  P2T19E: .ASCIZ  /OVERWRITE/
 3388  006504  042523  045505  052040  SKTMES: .ASCIZ  /SEEK TIMES /
 3389  006520  050123  047111  046104  SRTMES: .ASCIZ  /SPINDLE ROTATION TIME /
```

```
3390  006547     050  052123  052101  VALDES: .ASCIZ  /(STATED IN 100'S OF MICRO SEC)/
3391  006606  050101  051120  054117  MAPROX: .ASCIZ  /APPROX /
3392  006616  047111  042516  000122  LABIN:  .ASCIZ  /INNER/
3393  006624  044515  042104  042514  LABMID: .ASCIZ  /MIDDLE/
3394  006633     117  052125  051105  LABOUT: .ASCIZ  /OUTER/
3395  006641     105  050130  041505  LABEXP: .ASCIZ  /EXPECTED/
3396  006652  030060  020061  054503  LABOCF: .ASCIZ  /001 CYL FWD/
3397  006666  030060  020061  054503  LABOCR: .ASCIZ  /001 CYL REV/
3398  006702  031061  020070  054503  LABHCF: .ASCIZ  /128 CYL FWD/
3399  006716  031061  020070  054503  LABHCR: .ASCIZ  /128 CYL REV/
3400  006732  032462  020066  054503  LABACF: .ASCIZ  /256 CYL FWD/
3401  006746  032462  020066  054503  LABACR: .ASCIZ  /256 CYL REV/
3402  006762  042110  020123  040506  HDMOVF: .ASCIZ  /HDS FAILED TO MOVE IN 10 TRIES/
3420  007021     122  051505  052105  OPR12:  .ASCIZ  /RESET WRT LCK /
3421  007040  047117  000040          OPR1A:  .ASCIZ  /ON /
3422  007044  047117  042040  053122  OPR1B:  .ASCIZ  /ON DRV /
3423  007054  047125  042504  020122  UNDTST: .ASCIZ  /UNDER TEST/
3424  007067     123  052105  053440  OPR004: .ASCIZ  /SET WRT LCK /
3425  007104  044504  043106  000040  DIFWD:  .ASCIZ  /DIFF /
3426  007112  043523  020116     000  SGNWD:  .ASCIZ  /SGN /
3427  007117     110  020104     000  HDWD:   .ASCIZ  /HD /
3428  007123     123  041505  000040  SECWD:  .ASCIZ  /SEC /
3429  007130  054503  020114     000  CYLWD:  .ASCIZ  /CYL /
3430  007135     106  047522  020115  FRMWD:  .ASCIZ  /FROM /
3431  007143     040  054502  040520  BYPSNM: .ASCIZ  / BYPASSED /
3432  007156  047522  052125  047111  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ (IN SEQ CALLED): /
3433  007221     104  053122  051440  STAMES: .ASCIZ  /DRV STAT/
3434  007232  040502  020104  042523  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD.  ALL SEC ASSUMED GOOD./
3435  007310  047524  020124  047503  TCERR:  .ASCIZ  /TOT COMPARE ERRS: /
3436
3437                                  ;       RESULT NAMES
3438  007333     104  053122  051040  MDRDY:  .ASCIZ  /DRV RDY /
3439  007344  047503  052116  042440  MCERR:  .ASCIZ  /CONT ERR /
3440  007356  042110  020122  051103  MHCRC:  .ASCIZ  /HDR CRC/
3441  007366  040504  040524  041440  MDCRC:  .ASCIZ  /DATA CRC/
3442  007377     110  051104  047040  MHNF:   .ASCIZ  /HDR NOT FND/
3443  007413     104  052101  020101  MDLT:   .ASCIZ  /DATA LATE/
3444  007425     110  051104  047040  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
3445  007455     104  053122  042440  MDRERR: .ASCIZ  /DRV ERR /
3454  007466  051104  020126  042523  MDSERR: .ASCIZ  /DRV SEL ERR /
3455  007503     104  053122  051440  MDRVST: .ASCIZ  /DRV STATE /
3456  007516  050123  047111  052040  MSPERR: .ASCIZ  /SPIN TIMEOUT /
3457  007534  051127  020124  040507  MWGERR: .ASCIZ  /WRT GAT ERR /
3458  007551     123  042505  020113  MSTERR: .ASCIZ  /SEEK TIMEOUT /
3459  007567     110  040505  020104  MHCERR: .ASCIZ  /HEAD CUR ERR /
3460  007605     127  052122  042040  MWDERR: .ASCIZ  /WRT DAT ERR /
3461  007622  050117  044440  041516  MOPERR: .ASCIZ  /OP INCOMPLETE /
3462  007641     110  051104  042057  MHDERR: .ASCIZ  &HDR/DAT ERR &
3463  007656  042110  020122  047516  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
3464  007704  047516  026516  054105  MNEERR: .ASCIZ  /NON-EXSTNT MEM /
3465  007724  054503  020114     000  MCYLOC: .ASCIZ  /CYL /
3466  007731     103  052517  042114  MNDRST: .ASCIZ  /COULD NOT RETRIEVE DRIVE STATUS/
3467  007771     125  045516  020116  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
3468  010036  040506  046111  052040  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLEAR/
3469  010077     127  044522  042524  MWRTAB: .ASCIZ  /WRITE ABORTED/
3470  010115     040  051105  020122  MEXERS: .ASCIZ  / ERR LIMIT EXCEEDED - UNIT DROPPED/
```

```
3471  010160  042440  051122  051117  MERRS:  .ASCIZ  / ERROR/
3472  010167     207  177777     000  BELL:   .ASCIZ  <207><377><377>
3473
3474                                   ;       RESULT SETTINGS
3475  010173     111  020123     000  RESE3:  .ASCIZ  /IS /
3476  010177     040  041123  000040  RESE4:  .ASCIZ  / SB /
3477
3478                                   ;       RESULT CONDITIONS
3479  010204  044440  020116     000  RESE5:  .ASCIZ  / IN /
3480  010211     040  043117  000040  RESE6:  .ASCIZ  / OF /
3481  010216  052123  052101  020105  STATE2: .ASCIZ  /STATE 2/
3482  010226  052123  052101  020105  STATE3: .ASCIZ  /STATE 3/
3483  010236  052123  052101  020105  STATE5: .ASCIZ  /STATE 5/
3487  010246  044506  051522  020124  C10MS:  .ASCIZ  /FIRST 3 MS/
3488  010261     065  030060  051515  C500MS: .ASCIZ  /500MS/
3489  010267     103  041531  042514  CCYLUP: .ASCIZ  /CYCLE UP/
3490  010300  040504  040524  054040  CAFDT:  .ASCIZ  /DATA XFER/
3491  010312  020065  042523  042103  C5SEC:  .ASCIZ  /5 SECDS/
3492
3493  010322  047045  052045  047045  FMTOP1: .ASCIZ  /%N%T%N%T%T%06%S%T%01%N/
3494  010351     045  022516  022524  FMTOP2: .ASCIZ  /%N%T%01%S1%T%01%N/
3495  010373     045  022516  022524  FMTOP3: .ASCIZ  /%N%T%01%S1%T%T%N/
3496  010414  052045  052045     000  FMT1:   .ASCIZ  /%T%T/
3497  010421     045  022516  022524  FMT1.1: .ASCIZ  /%N%T%T/
3498  010430  052045     000          FMT2:   .ASCIZ  /%T/
3499  010433     045  000116          FMT3:   .ASCIZ  /%N/
3500  010436  047045  052045  052045  FMT4:   .ASCIZ  /%N%T%T%N/
3501  010447     045  022516  022524  FMT5:   .ASCIZ  /%N%T%06%S1%T%01/
3502  010467     045  022516  030523  FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
3503  010531     045  022516  022524  FMT7:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
3504  010601     045  022516  022524  FMT8:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
3505  010633     045  022516  000124  FMT9:   .ASCIZ  /%N%T/
3506  010640  052045  047445  000061  FMT11:  .ASCIZ  /%T%01/
3507  010646  052045  047445  000063  FMT12:  .ASCIZ  /%T%03/
3508  010654  047045  051445  030461  FMT13:  .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
3509  010720  047045  052045  052045  FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
3510  010752  047045  051445  030461  FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
3511  011006  047045  051445  022465  FMT16:  .ASCIZ  /%N%S5%06/
3512  011017     045  030523  022460  FMT17:  .ASCIZ  /%S10%T%N%S11%06%N/
3513  011041     045  022516  030523  FMT18:  .ASCIZ  /%N%S13%T%S5%T%S4%T%S5%T%N/
3514  011073     045  022524  031123  FMT19:  .ASCIZ  /%T%S2%D6%S4%D6%S4%D6%S4%D6%N/
3515  011130  052045  051445  022462  FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
3516  011160  052045  051445  031061  FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
3517  011203     045  022516  030523  FMT22:  .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
3518  011237     045  022524  022524  FMT23:  .ASCIZ  /%T%T%T%01%N/
3519  011253     045  022516  000124  FMT24:  .ASCIZ  /%N%T/
3520  011260  047045  042045  022462  FMT25:  .ASCIZ  /%N%D2%T/
3521  011270  047045  051445  022461  FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
3522  011314  047045  052045  042045  FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
3523  011333     045  022516  022524  FMT28:  .ASCIZ  /%N%T%T%T/
3524  011344          ENDMOD
3529
3530  011344          BGNMOD  GLBERR
3531                  ;       ERR1    R3 POINTS TO RESULT MESSAGE
3532                  ;               RESULT: (R3)
3533
```

```
                                        ;       ERR2    R3 POINTS TO RESULT NAME
                                        ;               RESULT: (R3) IS 1 SB 0
                                        ;       ERR3    R3 POINTS TO RESULT NAME
                                        ;               RESULT: (R3) IS 0 SB 1
                                        ;       ERR4    R3 POINTS TO RESULT NAME
                                        ;               R4 POINTS TO RESULT CONDITIONS
                                        ;               RESULT: (R3) IS 1 SB 0 (R4)
                                        ;       ERR5    R3 POINTS TO RESULT NAME
                                        ;               R4 POINTS TO RESULT CONDITIONS
                                        ;               RESULT: (R3) IS 0 SB 1 (R4)
                                        ;       ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
                                        ;               REPORTS ALL
                                        ;               RESULT: "ERROR" IS 1 SB 0
                                        ;       ERR7    DRIVE STATE ERROR REPORT
                                        ;               R3 CONTAINS EXPECTED STATE
                                        ;               T.STAT CONTAINS BAD STATE
                                        ;               RESULT: DRIVE STATE IS (T.STAT) SB (R3)
                                        ;       ERR8    HEAD POSITIONING ERROR REPORT
                                        ;               NEWCYL CONTAINS EXPECTED CYLINDER
                                        ;               HDWRD1 CONTAINS BAD CYLINDER
                                        ;               RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
                                        ;       ERR9    UTILITY RESULT REPORT
                                        ;               R3 POINTS TO RESULT NAME
                                        ;               R4 POINTS TO VALUE 1
                                        ;               R5 POINTS TO VALUE 2
                                        ;               RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
                                        ;       ERR10   COMPARE ERROR REPORT
                                        ;               R3 CONTAINS THE BAD WORD NUMBER
                                        ;               R4 POINTS TO BAD WORD
                                        ;               R5 POINTS TO GOOD WORD
                                        ;               RESULT: WORD (R3) IS (R4) SB (R5)

011344                                  BGNMSG  ERR1
011344  105737  002657                          TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
011350  001002                                  BNE     1$              ;YES - SKIP
011352  005237  002650                          INC     ERRCNT          ;ELSE BUMP ERROR COUNT
011356  010146                          1$:     MOV     R1,-(SP)        ;STORE R1
011360  004737  023000                          JSR     PC,RPTOP        ;REPORT OPERATION
011364  012721  000001                          MOV     #1,(R1)+        ;SET PARAM NUMBER
011370  010321                                  MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
011372  004737  023566                          JSR     PC,RPTRES       ;REPORT RESULTS
011376  004737  023774                          JSR     PC,RPTREM       ;REPORT REMAINDER
011402  012601                                  MOV     (SP)+,R1        ;RESTORE R1
011404  004737  014370                          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
011410                                  ENDMSG
(3) 011410                              L10000:
(3) 011410  104023                              EMT     C$MSG
```

```
3588
3589  011412                                BGNMSG  ERR2
3590  011412  005237  002650                        INC     ERRCNT          ;BUMP ERROR COUNT
3591  011416  010146                              MOV     R1,-(SP)        ;STORE R1
3592  011420  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3593  011424  012721  000003                      MOV     #3,(R1)+        ;SET PARAM NUMBER
3594  011430  010321                              MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3595  011432  012721  000001                      MOV     #1,(R1)+        ;SET IS VALUE
3596  011436  005021                              CLR     (R1)+           ;SET SB VALUE
3597  011440  004737  023566                      JSR     PC,RPTRES       ;REPORT RESULTS
3598  011444  004737  023774                      JSR     PC,RPTREM       ;REPORT REMAINDER
3599  011450  012601                              MOV     (SP)+,R1        ;RESTORE R1
3600  011452  004737  014370                      JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3601  011456                                ENDMSG
 (3)  011456                                L10001:
 (3)  011456  104023                              EMT     C$MSG
3602
3603  011460                                BGNMSG  ERR3
3604  011460  005237  002650                        INC     ERRCNT          ;BUMP ERROR COUNT
3605  011464  010146                              MOV     R1,-(SP)        ;STORE R1
3606  011466  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3607  011472  012721  000003                      MOV     #3,(R1)+        ;SET PARAM NUMBER
3608  011476  010321                              MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3609  011500  005021                              CLR     (R1)+           ;SET IS VALUE
3610  011502  012721  000001                      MOV     #1,(R1)+        ;SET SB VALUE
3611  011506  004737  023566                      JSR     PC,RPTRES       ;REPORT RESULTS
3612  011512  004737  023774                      JSR     PC,RPTREM       ;REPORT REMAINDER
3613  011516  012601                              MOV     (SP)+,R1        ;RESTORE R1
3614  011520  004737  014370                      JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3615  011524                                ENDMSG
 (3)  011524                                L10002:
 (3)  011524  104023                              EMT     C$MSG
3616
3617  011526                                BGNMSG  ERR4
3618  011526  005237  002650                        INC     ERRCNT          ;BUMP ERROR COUNT
3619  011532  010146                              MOV     R1,-(SP)        ;STORE R1
3620  011534  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3621  011540  012721  000004                      MOV     #4,(R1)+        ;SET PARAM NUMBER
3622  011544  010321                              MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3623  011546  012721  000001                      MOV     #1,(R1)+        ;SET IS VALUE
3624  011552  005021                              CLR     (R1)+           ;SET SB VALUE
3625  011554  010411                              MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
3626  011556  004737  023566                      JSR     PC,RPTRES       ;REPORT RESULTS
3627  011562  004737  023774                      JSR     PC,RPTREM       ;REPORT REMAINDER
3628  011566  012601                              MOV     (SP)+,R1        ;RESTORE R1
3629  011570  004737  014370                      JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3630  011574                                ENDMSG
 (3)  011574                                L10003:
 (3)  011574  104023                              EMT     C$MSG
3631
3632  011576                                BGNMSG  ERR5
3633  011576  005237  002650                        INC     ERRCNT          ;BUMP ERROR COUNT
3634  011602  010146                              MOV     R1,-(SP)        ;STORE R1
3635  011604  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3636  011610  012721  000004                      MOV     #4,(R1)+        ;SET PARAM NUMBER
3637  011614  010321                              MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
```

```
 3638  011616  005021                          CLR     (R1)+           ;SET IS VALUE
 3639  011620  012721  000001                  MOV     #1,(R1)+        ;SET SB VALUE
 3640  011624  010411                          MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
 3641  011626  004737  023566                  JSR     PC,RPTRES       ;REPORT RESULTS
 3642  011632  004737  023774                  JSR     PC,RPTREM       ;REPORT REMAINDER
 3643  011636  012601                          MOV     (SP)+,R1        ;RESTORE R1
 3644  011640  004737  014370                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 3645  011644                          ENDMSG
  (3)  011644                          L10004:
  (3)  011644  104023                          EMT     C$MSG
 3646
 3647  011646                          BGNMSG  ERR6
 3648  011646  105737  002657                  TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
 3649  011652  001002                          BNE     17$             ;YES - SKIP
 3650  011654  005237  002650                  INC     ERRCNT          ;ELSE BUMP ERROR COUNT
 3651  011660  010146                  17$:    MOV     R1,-(SP)        ;STORE R1
 3652  011662  010346                          MOV     R3,-(SP)        ;STORE R3
 3653  011664  010446                          MOV     R4,-(SP)        ;STORE R4
 3654  011666  010546                          MOV     R5,-(SP)        ;STORE R5
 3655  011670  004737  023000                  JSR     PC,RPTOP        ;REPORT OPERATION
 3656  011674  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
 3657  011700  012761  000001  000002          MOV     #1,2(R1)        ;INSERT IS VALUE
 3658  011706  005037  002536                  CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
 3659  011712  013703  002456                  MOV     T.CS,R3         ;GET T.CS
 3660  011716  042703  177761                  BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
 3661  011722  022703  000004                  CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
 3662  011726  001432                          BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
 3663  011730  012762  000003  000004          MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
 3664  011736  012703  000004                  MOV     #4,R3
 3665  011742  053703  002444                  BIS     RLDRV,R3
 3666  011746  010362  000000                  MOV     R3,RLCS(R2)
 3667  011752                          WAITUS  #10.            ;WAIT FOR CONTROLLER READY
  (3)  011752  012700  000012                  MOV     #10.,R0
  (3)  011756  104027                          EMT     C$WTU
 3668  011760  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
 3669  011766  001003                          BNE     10$             ;YES - SKIP
 3670  011770  012703  001000          9$:     MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
 3671  011774  000413                          BR      2$              ;IN MESSAGE WORD AND SKIP
 3672  011776  016203  000006          10$:    MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
 3673  012002  010337  002536                  MOV     R3,TEMP3
 3674  012006  113703  002537                  MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
 3675  012012  000402                          BR      13$
 3676  012014  113703  002465          1$:     MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
 3677  012020  042703  177442          13$:    BIC     #177442,R3      ;CLEAR UNUSED BITS
 3678  012024  013704  002456          2$:     MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
 3679  012030  042704  001777                  BIC     #1777,R4        ;CLEAR UNUSED BITS
 3680  012034  050403                          BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
 3681  012036  032703  002000                  BIT     #OPIERR,R3      ;TEST IF OPI SET
 3682  012042  001442                          BEQ     115$            ;NO - SKIP
 3683  012044  032703  010000                  BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
 3684  012050  001026                          BNE     107$            ;YES - SKIP
 3685  012052  032703  004000                  BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
 3686  012056  001020                          BNE     105$            ;YES - SKIP
 3687  012060  012704  007622                  MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
 3688  012064                          100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
 (10)  012064  012746  010160                  MOV     #MERRS,-(SP)
```

```
   (9) 012070 010446                        MOV     R4,-(SP)
   (8) 012072 012746 004747                 MOV     #MRSLT,-(SP)
   (7) 012076 012746 011333                 MOV     #FMT28,-(SP)
   (6) 012102 012746 000004                 MOV     #4,-(SP)
   (3) 012106 010600                        MOV     SP,R0
   (4) 012110 104014                        EMT     C$PNTB
   (4) 012112 062706 000012                 ADD     #12,SP
  3689 012116 000430                        BR      120$            ;SKIP
  3690 012120 012704 007356         105$:   MOV     #MHCRC,R4       ;HDR CRC MESSAGE
  3691 012124 000757                        BR      100$
  3692 012126 032703 004000         107$:   BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
  3693 012132 001003                        BNE     109$            ;YES - SKIP
  3694 012134 012704 007377                 MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
  3695 012140 000751                        BR      100$
  3696 012142 012704 007425         109$:   MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
  3697 012146 000746                        BR      100$            ;SKIP
  3698 012150 032703 004000         115$:   BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
  3699 012154 001403                        BEQ     118$            ;NO - SKIP
  3700 012156 012704 007366                 MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
  3701 012162 000740                        BR      100$            ;SKIP
  3702 012164 032703 010000         118$:   BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
  3703 012170 001403                        BEQ     120$            ;NO - SKIP
  3704 012172 012704 007413                 MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
  3705 012176 000732                        BR      100$            ;SKIP
  3706 012200 012705 100000         120$:   MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
  3707 012204 005004                        CLR     R4              ;CLEAR R4 FOR TABLE COUNT
  3708 012206 030503                3$:     BIT     R5,R3           ;TEST IF BIT IS SET
  3709 012210 001005                        BNE     6$              ;YES - SKIP TO REPORT
  3710 012212 005724                4$:     TST     (R4)+           ;ELSE BUMP TABLE POINTER
  3711 012214 000241                        CLC                     ;CLEAR CARRY
  3712 012216 006005                        ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
  3713 012220 001372                        BNE     3$              ;LOOP IF NOT 0
  3714 012222 000405                        BR      7$              ;ELSE REPORT REMAINDER
  3715 012224 016411 002164         6$:     MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
  3716 012230 004737 023566                 JSR     PC,RPTRES       ;REPORT RESULTS
  3717 012234 000766                        BR      4$              ;GET NEXT BIT
  3718 012236 004737 023774         7$:     JSR     PC,RPTREM       ;REPORT REMAINDER
  3719 012242 005737 002536                 TST     TEMP3           ;TEST IF ANY NEW STATUS
  3720 012246 001414                        BEQ     15$             ;NO - SKIP
  3721 012250                               PRINTB  #FMT17,#STAMES,TEMP3
   (9) 012250 013746 002536                 MOV     TEMP3,-(SP)
   (8) 012254 012746 007221                 MOV     #STAMES,-(SP)
   (7) 012260 012746 011017                 MOV     #FMT17,-(SP)
   (6) 012264 012746 000003                 MOV     #3,-(SP)
   (3) 012270 010600                        MOV     SP,R0
   (4) 012272 104014                        EMT     C$PNTB
   (4) 012274 062706 000010                 ADD     #10,SP
  3722 012300 032737 004000 002456  15$:    BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
  3723 012306 001453                        BEQ     25$             ;NO - SKIP
  3724 012310 032737 002000 002456          BIT     #OPIERR,T.CS    ;TEST IF OPI SET
  3725 012316 001047                        BNE     25$             ;YES - SKIP
  3726 012320 005037 002426                 CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
  3727 012324 012701 000200                 MOV     #128.,R1        ;SET COMPARE LENGTH
  3728 012330 012703 000001                 MOV     #1,R3           ;SET WORD COUNT
  3729 012334 012705 003656                 MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
  3730 012340 012704 003256                 MOV     #IBUFF,R4       ;SET TEST WORD POINTER
```

```
 3731  012344  021514                    18$:    CMP     (R5),(R4)       ;CHECK WORD
 3732  012346  001427                            BEQ     19$             ;GOOD - SKIP
 3733  012350  023727  002426  000012            CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
 3734  012356  003021                            BGT     20$             ;YES - SKIP
 3735  012360                                    PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
  (13) 012360  011546                            MOV     (R5),-(SP)
  (12) 012362  012746  010177                    MOV     #RESE4,-(SP)
  (11) 012366  011446                            MOV     (R4),-(SP)
  (10) 012370  012746  010173                    MOV     #RESE3,-(SP)
  (9)  012374  010346                            MOV     R3,-(SP)
  (8)  012376  012746  005575                    MOV     #MWORD,-(SP)
  (7)  012402  012746  010752                    MOV     #FMT15,-(SP)
  (6)  012406  012746  000007                    MOV     #7,-(SP)
  (3)  012412  010600                            MOV     SP,R0
  (4)  012414  104014                            EMT     C$PNTB
  (4)  012416  062706  000020                    ADD     #20,SP
 3736  012422  005237  002426            20$:    INC     MORECE          ;BUMP ERROR COUNTER
 3737  012426  022524                    19$:    CMP     (R5)+,(R4)+     ;BUMP POINTERS
 3738  012430  005203                            INC     R3              ;BUMP COUNTER
 3739  012432  005301                            DEC     R1              ;DEC LENGTH COUNT
 3740  012434  001343                            BNE     18$             ;LOOP IF NOT DONE
 3741  012436  005737  002426            25$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
 3742  012442  001421                            BEQ     27$             ;NO - SKIP
 3743  012444  012701  000200                    MOV     #128.,R1        ;SET COMPARE LENGTH
 3744  012450                                    PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
  (11) 012450  010146                            MOV     R1,-(SP)
  (10) 012452  012746  010211                    MOV     #RESE6,-(SP)
  (9)  012456  013746  002426                    MOV     MORECE,-(SP)
  (8)  012462  012746  007310                    MOV     #TCERR,-(SP)
  (7)  012466  012746  011314                    MOV     #FMT27,-(SP)
  (6)  012472  012746  000005                    MOV     #5,-(SP)
  (3)  012476  010600                            MOV     SP,R0
  (4)  012500  104014                            EMT     C$PNTB
  (4)  012502  062706  000014                    ADD     #14,SP
 3745  012506  012605                    27$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
 3746  012510  012604                            MOV     (SP)+,R4
 3747  012512  012603                            MOV     (SP)+,R3
 3748  012514  012601                            MOV     (SP)+,R1
 3749  012516  004737  014370                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 3750  012522                            ENDMSG
  (3)  012522                            L10005:
  (3)  012522  104023                            EMT     C$MSG
 3751
 3752  012524                            BGNMSG  ERR7
 3753  012524  005237  002650                    INC     ERRCNT          ;BUMP ERROR COUNT
 3754  012530  010146                            MOV     R1,-(SP)        ;STORE R1
 3755  012532  004737  023000                    JSR     PC,RPTOP        ;REPORT OPERATION
 3756  012536  012721  000003                    MOV     #3,(R1)+        ;SET PARAM NUMBER
 3757  012542  012721  007503                    MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
 3758  012546  013721  002472                    MOV     T.STAT,(R1)+    ;INSERT IS VALUE
 3759  012552  010311                            MOV     R3,(R1) ;INSERT SB VALUE
 3760  012554  004737  023566                    JSR     PC,RPTRES       ;REPORT RESULTS
 3761  012560  004737  023774                    JSR     PC,RPTREM       ;REPORT REMAINDER
 3762  012564  012601                            MOV     (SP)+,R1        ;RESTORE R1
 3763  012566  004737  014370                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 3764  012572                            ENDMSG
```

```
  (3)  012572                              L10006:
  (3)  012572  104023                              EMT     C$MSG
3765
3766  012574                              BGNMSG  ERR8
3767  012574  005237  002650                      INC     ERRCNT          ;BUMP ERROR COUNT
3768  012600  010146                              MOV     R1,-(SP)        ;STORE R1
3769  012602  010346                              MOV     R3,-(SP)        ;STORE R3
3770  012604  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3771  012610  012721  000003                      MOV     #3,(R1)+        ;SET PARAM NUMBER
3772  012614  012721  007724                      MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
3773  012620  013711  002464                      MOV     HDWRD1,(R1)     ;GET HEADER WORD
3774  012624  012703  000007                      MOV     #7,R3           ;SET SHIFT COUNT
3775  012630  000241                      3$:     CLC
3776  012632  006011                              ROR     (R1)            ;ALIGN CHAR FOR PRINTING
3777  012634  005303                              DEC     R3              ;  AS IS VALUE
3778  012636  001374                              BNE     3$
3779  012640  005721                              TST     (R1)+           ;BUMP PARAM POINTER
3780  012642  013711  002514                      MOV     NEWCYL,(R1)     ;INSERT SB VALUE
3781  012646  004737  023566                      JSR     PC,RPTRES       ;REPORT RESULTS
3782  012652  004737  023774                      JSR     PC,RPTREM       ;REPORT REMAINDER
3783  012656  012603                              MOV     (SP)+,R3        ;RESTORE R3
3784  012660  012601                              MOV     (SP)+,R1        ;RESTORE R1
3785  012662  004737  014370                      JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3786  012666                              ENDMSG
  (3)  012666                              L10007:
  (3)  012666  104023                              EMT     C$MSG
3787
3788  012670                              BGNMSG  ERR9
3789  012670  005237  002650                      INC     ERRCNT          ;BUMP ERROR COUNT
3790  012674  010146                              MOV     R1,-(SP)        ;STORE R1
3791  012676  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3792  012702  012721  000003                      MOV     #3,(R1)+        ;SET PARAM NUMBER
3793  012706  010321                              MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3794  012710  010421                              MOV     R4,(R1)+        ;SET IS VALUE
3795  012712  010521                              MOV     R5,(R1)+        ;SET SB VALUE
3796  012714  004737  023566                      JSR     PC,RPTRES       ;REPORT RESULTS
3797  012720  004737  023774                      JSR     PC,RPTREM       ;REPORT REMAINDER
3798  012724  012601                              MOV     (SP)+,R1        ;RESTORE R1
3799  012726  004737  014370                      JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3800  012732                              ENDMSG
  (3)  012732                              L10010:
  (3)  012732  104023                              EMT     C$MSG
3801  012734                              BGNMSG  ERR10
3802  012734  010146                              MOV     R1,-(SP)        ;STORE R1
3803  012736  005737  002426                      TST     MORECE          ;TEST IF 2ND BAD LINE
3804  012742  001051                              BNE     3$              ;YES - SKIP
3805  012744  005237  002650                      INC     ERRCNT          ;BUMP ERROR COUNT
3806  012750  004737  023000                      JSR     PC,RPTOP        ;REPORT OPERATION
3807  012754                                      PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
 (11) 012754  005046                              CLR     -(SP)
 (11) 012756  153716  002445                      BISB    RLDRV+1,(SP)
 (10) 012762  012746  005423                      MOV     #DRVNAM,-(SP)
 (9)  012766  013746  002440                      MOV     RLBAS,-(SP)
 (8)  012772  012746  005412                      MOV     #BASADD,-(SP)
 (7)  012776  012746  010447                      MOV     #FMT5,-(SP)
 (6)  013002  012746  000005                      MOV     #5,-(SP)
```

```
  (3)  013006  010600                        MOV     SP,R0
  (4)  013010  104014                        EMT     C$PNTB
  (4)  013012  062706  000014                ADD     #14,SP
3808   013016                                PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#PESE4,(R5)
 (14)  013016  011546                        MOV     (R5),-(SP)
 (13)  013020  012746  010177                MOV     #RESE4,-(SP)
 (12)  013024  011446                        MOV     (R4),-(SP)
 (11)  013026  012746  010173                MOV     #RESE3,-(SP)
 (10)  013032  010346                        MOV     R3,-(SP)
  (9)  013034  012746  005575                MOV     #MWORD,-(SP)
  (8)  013040  012746  004747                MOV     #MRSLT,-(SP)
  (7)  013044  012746  010720                MOV     #FMT14,-(SP)
  (6)  013050  012746  000010                MOV     #10,-(SP)
  (3)  013054  010600                        MOV     SP,R0
  (4)  013056  104014                        EMT     C$PNTB
  (4)  013060  062706  000022                ADD     #22,SP
3809   013064  000421                        BR      4$
3810   013066                          3$:   PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
 (13)  013066  011546                        MOV     (R5),-(SP)
 (12)  013070  012746  010177                MOV     #RESE4,-(SP)
 (11)  013074  011446                        MOV     (R4),-(SP)
 (10)  013076  012746  010173                MOV     #RESE3,-(SP)
  (9)  013102  010346                        MOV     R3,-(SP)
  (8)  013104  012746  005575                MOV     #MWORD,-(SP)
  (7)  013110  012746  010752                MOV     #FMT15,-(SP)
  (6)  013114  012746  000007                MOV     #7,-(SP)
  (3)  013120  010600                        MOV     SP,R0
  (4)  013122  104014                        EMT     C$PNTB
  (4)  013124  062706  000020                ADD     #20,SP
3811   013130  005237  002426          4$:   INC     MORECE          ;INC COMPARE ERROR COUNT
3812   013134  012601                        MOV     (SP)+,R1        ;RESTORE R1
3813   013136  004737  014370                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3814   013142                          ENDMSG
 (3)   013142                          L10011:
 (3)   013142  104023                        EMT     C$1SG
3815   013144                          ENDMOD
3816                                          .EVEN
3817
3818   013144                          BGNMOD  HPTCODE
3819   013144                          BGNHW
 (3)   013144  000005                        .WORD   L10012-L$HW/2
3820   013146  174400                        .WORD   174400          ;CSR BASE ADDRESS DEFAULT
3821   013150  000330                        .WORD   330             ;VECTOR DEFAULT
3822   013152  000240                        .WORD   240             ;PRIORITY DEFAULT
3823   013154  000000                        .WORD   0               ;DRIVE NUMBER DEFAULT
3824   013156  000001                        .WORD   1               ;RL11 CONTROLLER
3825   013160                          ENDHW
 (3)   013160                          L10012:
3826   013160                          ENDMOD
3827
3828   013160                          BGNMOD  SPTCODE
3829   013160                          BGNSW
 (3)   013160  000006                        .WORD   L10013-L$SW/2
3830   013162  000000                  MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
3831                                                                   ;BIT 1 = USE ALL SECTORS
3832                                                                   ;BIT 2 = EXECUTE DRIVE SELECT TEST
```

```
                                                        ;BIT 3 = EXECUTE HEAD ALIGNMENT
                                                        ;BIT 4 = DROP DRIVE IF NO RESPONSE
                                                        ;BIT 12 = HEAD SELECT SUPPLIED FLAG
                                                        ;BIT 13 = HILIMIT SPECIFIED FLAG
                                                        ;BIT 14 = LO LIMIT SPECIFIED FLAG
                                                        ;BIT 15 = DO MANUAL INTERVENTION
013164  000000                  LOLIMW: .WORD   0
013166  000377                  HILIMW: .WORD   255.
013170  000000                  HEADW:  .WORD   0
013172  000024                  ERLIMW: .WORD   20.              ;ERROR LIMIT
013174  000012                  DCLIMW: .WORD   10.              ;COMPARE ERROR LIMIT
013176                          ENDSW
013176                          L10013:
013176                          ENDMOD

013176                          BGNMOD  DSPCODE
013176                          DISPATCH        19
013176  000023                          .WORD   19
013200  024260                          .WORD   T1
013202  024602                          .WORD   T2
013204  025072                          .WORD   T3
013206  025304                          .WORD   T4
013210  025514                          .WORD   T5
013212  025724                          .WORD   T6
013214  026150                          .WORD   T7
013216  026350                          .WORD   T8
013220  026576                          .WORD   T9
013222  027072                          .WORD   T10
013224  027370                          .WORD   T11
013226  027666                          .WORD   T12
013230  031434                          .WORD   T13
013232  032144                          .WORD   T14
013234  032360                          .WORD   T15
013236  033104                          .WORD   T16
013240  034126                          .WORD   T17
013242  035120                          .WORD   T18
013244  036230                          .WORD   T19
013246                          ENDMOD

013246                          BGNMOD  INITCODE
013246                          BGNINIT
013246                                  SETPRI  #340
013246  012700  000340                  MOV     #340,R0
013252  104041                          EMT     C$SPRI
013254                                  MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
013254  104051                          EMT     C$MANI
013256                                  BCOMPLETE       1$      ;YES - SKIP
013256  103403                          BCS     1$
013260  042737  100014  013162          BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
                                                                ; INTERVENTION FLAGS
013266  005037  002414          1$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
013272                                  READEF  #EF.PWR         ;POWER FAILURE
013272  012700  000034                  MOV     #EF.PWR,R0
013276  104050                          EMT     C$REFG
013300                                  BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
013300  103004                          BCC     4$
```

```
 3866  013302  013737  002014  002662          MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
 3867  013310  000513                          BR      PWCON           ;GO SERVICE POWER FAIL
 3868  013312                          4$:     READEF  #EF.START       ;CHECK IF START
  (3)  013312  012700  000040                  MOV     #EF.START,R0
  (3)  013316  104050                          EMT     C$REFG
 3869  013320                                  BNCOMPLETE      RESTART ;NO - SKIP
  (2)  013320  103031                          BCC     RESTART
 3870                                  ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
 3871                                  ;       PASS COUNT, AND ERROR COUNT.
 3872  013322  013737  002014  002506          MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
 3873  013330  005037  002652          RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
 3874  013334  012737  177777  002654          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
 3875  013342  012737  177777  002422          MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
 3876  013350  032737  020000  013162          BIT     #HICYL,MISWIW   ;TEST IF HI LIMIT SET
 3877  013356  001003                          BNE     3$              ;YES - SKIP
 3878  013360  012737  000377  013166          MOV     #377,HILIMW     ;ELSE INIT HILIMIT
 3879  013366  032737  040000  013162  3$:     BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
 3880  013374  001002                          BNE     5$              ;YES - SKIP
 3881  013376  005037  013164                  CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
 3882  013402  000427                  5$:     BR      SETDON
 3883  013404                          RESTART:
 3884  013404                                  READEF  #EF.RESTART     ;CHECK IF RESTART
  (3)  013404  012700  000037                  MOV     #EF.RESTART,R0
  (3)  013410  104050                          EMT     C$REFG
 3885  013412                                  BCOMPLETE       RSTRT   ;NO - SKIP
  (2)  013412  103746                          BCS     RSTRT
 3886  013414                          CONTINUE:
 3887  013414                                  READEF  #EF.CONTINUE    ;TEST IF CONTINUE
  (3)  013414  012700  000036                  MOV     #EF.CONTINUE,R0
  (3)  013420  104050                          EMT     C$REFG
 3888  013422                                  BCOMPLETE       PWCON
  (2)  013422  103446                          BCS     PWCON
 3889                                  ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST
 3890  013424                                  READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
  (3)  013424  012700  000035                  MOV     #EF.NEW,R0
  (3)  013430  104050                          EMT     C$REFG
 3891  013432                                  BCOMPLETE       PASNEW
  (2)  013432  103403                          BCS     PASNEW
 3892  013434                          NXTPAS:
 3893  013434  005737  002506                  TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
 3894  013440  001010                          BNE     SETDON          ;NO - SKIP
 3895  013442  005237  002652          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
 3896  013446  013737  002014  002506          MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
 3897  013454  012737  177777  002654          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
 3898  013462  005237  002654          SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
 3899  013466  005337  002506                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
 3900  013472  013700  002654                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
 3901  013476  012702  002440                  MOV     #RLBAS,R2
 3902  013502                                  GPHARD  R0,R1
  (3)  013502  104042                          EMT     C$GPHRD
  (3)  013504  010001                          MOV     R0,R1
 3903  013506                                  BCOMPLETE       7$      ;SKIP IF GOOD PARAM
  (2)  013506  103406                          BCS     7$
 3904  013510  005737  002662                  TST     PWRFLG          ;RECENT POWER FAILURE
 3905  013514  001747                          BEQ     NXTPAS  ;NO
 3906  013516  005337  002662                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
```

```
 3907  013522  000744                          BR      NXTPAS
 3908  013524  012122                  7$:     MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
 3909  013526  012122                          MOV     (R1)+,(R2)+     ;                    VECTOR
 3910  013530  005721                          TST     (R1)+           ;BUMP PAST PRIORITY
 3911  013532  012122                          MOV     (R1)+,(R2)+     ;                    DRIVE
 3912  013534  005037  002650                  CLR     ERRCNT                  ;CLEAR OUT ERROR COUNT
 3913  013540                          PWCON:  SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
  (7)  013540  012746  000340                  MOV     #340,-(SP)
  (6)  013544  012746  014332                  MOV     #INTHLR,-(SP)
  (5)  013550  013746  002442                  MOV     RLVEC,-(SP)
  (4)  013554  012746  000003                  MOV     #3,-(SP)
  (3)  013560  104037                          EMT     C$SVEC
  (2)  013562  062706  000010                  ADD     #10,SP
 3914  013566                                  SETPRI  #0              ;SET PRIORITY
  (3)  013566  012700  000000                  MOV     #0,R0
  (3)  013572  104041                          EMT     C$SPRI
 3915  013574  013702  002440                  MOV     RLBAS,R2        ;SET RL BASE ADDRESS POINTER
 3916
 3917
 3918
 3919                                  ;       CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
 3920                                  ;       ERROR SETS ON GET STATUS.
 3921  013600  005737  002652                  TST     PASNUM          ;TEST IF PASS 0
 3922  013604  001135                          BNE     22$             ;NO - SKIP
 3923  013606  032737  000020  013162          BIT     #AUTOSZ,MISWIW  ;TEST IF DOING AUTO SIZE
 3924  013614  001531                          BEQ     22$             ;NO - SKIP
 3925                                  ;CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
 3926  013616  005037  002660                  CLR     TRPFLG          ;TRAP OCCURANCE
 3927  013622                                  SETVEC  ERRVEC,#TRPHAN,#340     ;SET TRAP VECTOR
  (7)  013622  012746  000340                  MOV     #340,-(SP)
  (6)  013626  012746  014324                  MOV     #TRPHAN,-(SP)
  (5)  013632  013746  002642                  MOV     ERRVEC,-(SP)
  (4)  013636  012746  000003                  MOV     #3,-(SP)
  (3)  013642  104037                          EMT     C$SVEC
  (2)  013644  062706  000010                  ADD     #10,SP
 3928  013650  005762  000000                  TST     RLCS(R2)        ;ACCESS BUS
 3929  013654  005737  002660                  TST     TRPFLG          ;TRAP OCCUR??
 3930  013660  001032                          BNE     5$              ;YES, DON'T INVESTIGATE FURTHER
 3931  013662  013705  002444                  MOV     RLDRV,R5        ;GET DRIVE NUMBER
 3932  013666  052705  000200                  BIS     #CRDYMSK,R5     ;INSERT CONT READY
 3933  013672  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD IN DRIVE NUMBER
 3934  013676  032762  000001  000000          BIT     #DRDYMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
 3935  013704  001072                          BNE     20$             ;YES - GO DO TEST
 3936  013706  012762  000003  000004          MOV     #GETSTAT,RLDA(R2) ;ELSE INSERT GET STATUS
 3937  013714  052705  000004                  BIS     #4,R5           ;LOAD R5 WITH GET STATUS FUNCTION
 3938  013720  042705  000200                  BIC     #CRDYMSK,R5     ;CLEAR CONTROLLER READY
 3939  013724  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD CS REG
 3940  013730                          WAITMS  #4                      ;WAIT 4 MS
  (3)  013730  012700  000004                  MOV     #4,R0
  (3)  013734  104026                          EMT     C$WTM
 3941  013736  032762  002000  000000          BIT     #OPIERR,RLCS(R2);TEST IF OPI SET
 3942  013744  001452                          BEQ     20$             ;NO - SKIP
 3943  013746                          5$:     CLRVEC  ERRVEC
  (3)  013746  013700  002642                  MOV     ERRVEC,R0
  (3)  013752  104036                          EMT     C$CVEC
 3944  013754                                  PRINTF  #FMT24,#DRVNAV
```

```
  (8)  013754  012746  005430                    MOV     #DRVNAV,-(SP)
  (7)  013760  012746  011253                    MOV     #FMT24,-(SP)
  (6)  013764  012746  000002                    MOV     #2,-(SP)
  (3)  013770  010600                            MOV     SP,R0
  (4)  013772  104017                            EMT     C$PNTF
  (4)  013774  062706  000006                    ADD     #6,SP
3945   014000                          10$:       PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  014000  005046                            CLR     -(SP)
 (11)  014002  153716  002445                    BISB    RLDRV+1,(SP)
 (10)  014006  012746  005423                    MOV     #DRVNAM,-(SP)
  (9)  014012  013746  002440                    MOV     RLBAS,-(SP)
  (8)  014016  012746  005412                    MOV     #BASADD,-(SP)
  (7)  014022  012746  010447                    MOV     #FMT5,-(SP)
  (6)  014026  012746  000005                    MOV     #5,-(SP)
  (3)  014032  010600                            MOV     SP,R0
  (4)  014034  104017                            EMT     C$PNTF
  (4)  014036  062706  000014                    ADD     #14,SP
3946   014042                                    PRINTF  #FMT3
  (7)  014042  012746  010433                    MOV     #FMT3,-(SP)
  (6)  014046  012746  000001                    MOV     #1,-(SP)
  (3)  014052  010600                            MOV     SP,R0
  (4)  014054  104017                            EMT     C$PNTF
  (4)  014056  062706  000004                    ADD     #4,SP
3947   014062                                    DODU    PSETNM          ;DROP DRIVE
  (3)  014062  013700  002654                    MOV     PSETNM,R0
  (3)  014066  104053                            EMT     C$DODU
3948   014070                                    DOCLN
  (3)  014070  104044                            EMT     C$DCLN
3949   014072                          20$:       CLRVEC  ERRVEC
  (3)  014072  013700  002642                    MOV     ERRVEC,R0
  (3)  014076  104036                            EMT     C$CVEC
3950   014100                          22$:
3961                                   ;CHECK IF POWER FAILURE WAIT IS NEEDED
3962
3963   014100  005737  002662          4$:       TST     PWRFLG          ;NEEDED???
3964   014104  001434                            BEQ     8$              ;NO, SKIP
3965
3966   014106  013705  002444                    MOV     RLDRV,R5        ;DRIVE SELECT
3967   014112  052705  000200                    BIS     #CRDYMSK,R5     ;SET CRDY
3968   014116  010562  000000                    MOV     R5,RLCS(R2)     ;SELECT DRIVE
3969   014122  012701  000074                    MOV     #60.,R1         ;SIXTY SECOND TIMER
3970   014126  032762  000001  000000  9$:       BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET
3971   014134  001020                            BNE     8$              ;YES START TEST
3972
3973   014136                                    WAITMS  #10.            ;WAIT A SECOND
  (3)  014136  012700  000012                    MOV     #10.,R0
  (3)  014142  104026                            EMT     C$WTM
3974   014144  005301                            DEC     R1              ;SIXTY GONE BY
3975   014146  001367                            BNE     9$              ;NO
3976   014150                                    PRINTF  #FMT24,#NOPWR
  (8)  014150  012746  005463                    MOV     #NOPWR,-(SP)
  (7)  014154  012746  011253                    MOV     #FMT24,-(SP)
  (6)  014160  012746  000002                    MOV     #2,-(SP)
  (3)  014164  010600                            MOV     SP,R0
  (4)  014166  104017                            EMT     C$PNTF
  (4)  014170  062706  000006                    ADD     #6,SP
```

```
 3977  014174  000701                          BR      10$
 3978
 3979  014176                          8$:
 3980
 3981  014176                          ENDINIT
  (3)  014176                          L10014:
  (3)  014176  104011                          EMT     C$INIT
 3982  014200                          ENDMOD
 3983
 3984  014200                          BGNMOD  CLNCODE
 3985  014200                          BGNCLN
 3986
 3987  014200                                  SETVEC  ERRVEC,#TRPHAN,#340
  (7)  014200  012746  000340                  MOV     #340,-(SP)
  (6)  014204  012746  014324                  MOV     #TRPHAN,-(SP)
  (5)  014210  013746  002642                  MOV     ERRVEC,-(SP)
  (4)  014214  012746  000003                  MOV     #3,-(SP)
  (3)  014220  104037                          EMT     C$SVEC
  (2)  014222  062706  000010                  ADD     #10,SP
 3988
 3989  014226                                  SETPRI #7               ;SET PRORITY TO 7
  (3)  014226  012700  000007                  MOV     #7,R0
  (3)  014232  104041                          EMT     C$SPRI
 3990  014234  032762  000200  000000  2$:      BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
 3991  014242  001407                          BEQ     3$              ;NO LOOP UNTIL READY
 3992  014244  053762  002444  000000          BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
 3993  014252  032762  000001  000000          BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
 3994  014260  001003                          BNE     5$              ;NO - SKIP
 3995  014262                          3$:     WAITMS  #3              ;WAIT 300 MS
  (3)  014262  012700  000003                  MOV     #3,R0
  (3)  014266  104026                          EMT     C$WTM
 3996  014270                          5$:     CLRVEC  RLVEC           ;RELEASE VEC
  (3)  014270  013700  002442                  MOV     RLVEC,R0
  (3)  014274  104036                          EMT     C$CVEC
 3997  014276  005737  002662                  TST     PWRFLG  ;PWR FAIL SET
 3998  014302  001402                          BEQ     7$              ;NO
 3999  014304  005337  002662                  DEC     PWRFLG
 4000  014310                          7$:     CLRVEC  ERRVEC
  (3)  014310  013700  002642                  MOV     ERRVEC,R0
  (3)  014314  104036                          EMT     C$CVEC
 4001  014316                          ENDCLN
  (3)  014316                          L10015:
  (3)  014316  104012                          EMT     C$CLEAN
 4002
 4003  014320                          BGNDU
 4004  014320  000240                          NOP
 4005  014322                          ENDDU
  (3)  014322                          L10016:
  (3)  014322  104055                          EMT     C$DU
 4006
 4007  014324                          ENDMOD
 4008  014324                          BGNMOD  GLBSUB
 4009
 4010  014324  005237  002660          TRPHAN: INC     TRPFLG
 4011  014330  000002                          RTI
 4012
```

```
 4013  014332                                BGNSRV  INTHLR
 4014                                        ;       INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
 4015  014332                                        ABORTWAIT
  (3)  014332  104021                               EMT     C$ABRT
 4016  014334  012237  002456                       MOV     (R2)+,T.CS      ;STORE RL REGISTERS
 4017  014340  012237  002460                       MOV     (R2)+,T.BA
 4018  014344  012237  002462                       MOV     (R2)+,T.DA
 4019  014350  011237  002464                       MOV     (R2),T.MP
 4020  014354  012737  177777  002420               MOV     #-1,DONE        ;SET DONE FLAG
 4021  014362  013702  002440                       MOV     RLBAS,R2        ;RESTORE R2
 4022  014366                                ENDSRV
  (3)  014366                                L10017:
  (2)  014366  000002                               RTI
 4023
 4024                                        ;       ERROR LIMIT CHECKING ROUTINE
 4025                                        ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
 4026  014370  023737  002650  013172  CKERLM: CMP     ERRCNT,ERLIMW   ;TEST IF ERROR LIMIT EXCEEDED
 4027  014376  002453                           BLT     1$              ;NO - SKIP
 4028  014400                                    INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  014400  104020                           EMT     C$INLP
 4029  014402                                    BCOMPLETE       1$      ;YES - SKIP
  (2)  014402  103451                           BCS     1$
 4030  014404                                    PRINTF  #FMT25,ERLIMW,#MEXERS
  (9)  014404  012746  010115                   MOV     #MEXERS,-(SP)
  (8)  014410  013746  013172                   MOV     ERLIMW,-(SP)
  (7)  014414  012746  011260                   MOV     #FMT25,-(SP)
  (6)  014420  012746  000003                   MOV     #3,-(SP)
  (3)  014424  010600                           MOV     SP,R0
  (4)  014426  104017                           EMT     C$PNTF
  (4)  014430  062706  000010                   ADD     #10,SP
 4031  014434                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  014434  005046                           CLR     -(SP)
 (11)  014436  153716  002445                   BISB    RLDRV+1,(SP)
 (10)  014442  012746  005423                   MOV     #DRVNAM,-(SP)
  (9)  014446  013746  002440                   MOV     RLBAS,-(SP)
  (8)  014452  012746  005412                   MOV     #BASADD,-(SP)
  (7)  014456  012746  010447                   MOV     #FMT5,-(SP)
  (6)  014462  012746  000005                   MOV     #5,-(SP)
  (3)  014466  010600                           MOV     SP,R0
  (4)  014470  104017                           EMT     C$PNTF
  (4)  014472  062706  000014                   ADD     #14,SP
 4032  014476                                    PRINTF  #FMT3
  (7)  014476  012746  010433                   MOV     #FMT3,-(SP)
  (6)  014502  012746  000001                   MOV     #1,-(SP)
  (3)  014506  010600                           MOV     SP,R0
  (4)  014510  104017                           EMT     C$PNTF
  (4)  014512  062706  000004                   ADD     #4,SP
 4033  014516                                    DODU    PSETNM          ;DROP DRIVE
  (3)  014516  013700  002654                   MOV     PSETNM,R0
  (3)  014522  104053                           EMT     C$DODU
 4034  014524                                    DOCLN                   ;GO TO CLEAN UP
  (3)  014524  104044                           EMT     C$DCLN
 4035  014526  000207                    1$:     RTS     PC
 4036
 4037                                        ;       READ AND STORE ALL RL11 REGISTERS
 4038  014530  016237  000000  002456  READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
```

```
 4039  014536  016237  000002  002460          MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
 4040  014544  016237  000004  002462          MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
 4041  014552  016237  000006  002464          MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
 4042  014560  000207                          RTS     PC              ;RETURN
 4043
 4044                                  ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
 4045  014562  011646                  WAITIN: MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
 4046  014564  005066  000002                  CLR     2(SP)           ;CLEAR FOR POINTER
 4047  014570  032762  000200  000000          BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
 4048  014576  001420                          BEQ     4$              ;NO - SKIP TO WAIT
 4049  014600  004737  014530                  JSR     PC,READRL       ;READ ALL RL REGS
 4050  014604  005737  002420                  TST     DONE            ;TEST IF INTERRUPT OCCURRED
 4051  014610  001433                          BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
 4052  014612  012766  005603  000002  1$:     MOV     #MTOSLOW,2(SP)  ;ELSE SET TO SLOW ERROR POINTER
 4053  014620  032737  002000  002456          BIT     #OPIERR,T.CS    ;TEST IF OPI SET
 4054  014626  001403                          BEQ     2$              ;NO - SKIP
 4055  014630  012766  005622  000002          MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
 4056  014636  000207                  2$:     RTS     PC              ;RETURN
 4057  014640                          4$:     WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
  (3)  014640  012700  000003                  MOV     #3,R0
  (3)  014644  104026                          EMT     C$WTM
 4058  014646  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
 4059  014654  001006                          BNE     3$              ;YES - SKIP
 4060  014656  004737  014530                  JSR     PC,READRL       ;READ RL REGS
 4061  014662  012766  005705  000002          MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
 4062  014670  000762                          BR      2$              ;SKIP
 4063  014672  005737  002420          3$:     TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
 4064  014676  001345                          BNE     1$              ;YES - SKIP TO SET TO SLOW
 4065  014700  004737  014530          5$:     JSR     PC,READRL       ;READ RL REGS
 4066  014704  012766  005652  000002          MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
 4067  014712  000751                          BR      2$              ;GO TO RETURN
 4068
 4069                                  ;       OPERATION AND TEST INITIALIZE ROUTINE
 4070  014714  005037  002416          TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
 4071  014720  105037  002657                  CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
 4072  014724  005037  002426                  CLR     MORECE          ;RESET MORE COMPARE ERRORS
 4073  014730  000207                          RTS     PC
 4074
 4075                                  ;       GET STATUS AND GET STATUS WITH RESET ROUTINE
 4076  014732  013746  002540          GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
 4077  014736  012737  000013  002540          MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
 4078  014744  000412                          BR      GSTATG
 4079  014746  013746  002540          GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
 4080  014752  012737  000003  002540          MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
 4081  014760  000404                          BR      GSTATG
 4082  014762  013746  002540          GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
 4083  014766  005037  002540                  CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
 4084  014772  010346                  GSTATG: MOV     R3,-(SP)        ;STORE R3
 4085  014774  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4086  015000  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4087  015002  016663  000004  002250          MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
 4088  015010  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4089  015016  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4090  015022  010046                          MOV     R0,-(SP)        ;STORE R0
 4091  015024  010146                          MOV     R1,-(SP)        ;STORE R1
 4092  015026  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
```

```
 4093  015034  032737  000010  002540          BIT     #DRSET,TEMP4      ;TEST IF DRIVE RESET
 4094  015042  001453                          BEQ     11$               ;NO - SKIP
 4095  015044  032762  040000  000000          BIT     #DRVERR,RLCS(R2)  ;TEST IF DRIVE ERROR SET
 4096  015052  001403                          BEQ     49$               ;NO - SKIP
 4097  015054                                  WAITMS  #3                ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
   (3) 015054  012700  000003                  MOV     #3,R0
   (3) 015060  104026                          EMT     C$WTM
 4098  015062  012701  000062          49$:    MOV     #50.,R1           ;SET WAIT FOR 5 SEC
 4099  015066  004737  014762          50$:    JSR     PC,GSTAT          ;GET DRIVE STATUS
 4100  015072  015526                          3$
 4101  015074  032737  000001  002456          BIT     #DRDYMSK,T.CS     ;TEST IF DRIVE READY
 4102  015102  001051                          BNE     5$                ;YES - GO DO CLEAR
 4103  015104  032737  000020  002464          BIT     #HOSTAT,T.MP      ;ELSE TEST IF HEADS OUT
 4104  015112  001010                          BNE     51$               ;YES - BYPASS RELOAD WAIT FLAG SETTING
 4105  015114  032737  144000  002464          BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
 4106                                                                    ;THAT CAUSED HEADS TO
 4107                                                                    ;UNLOAD
 4108  015122  001441                          BEQ     5$                ;NO - SKIP
 4109  015124  052737  040000  002416          BIS     #RELDWT,OPFLAG    ;ELSE SET WAIT FLAG
 4110  015132  000435                          BR      5$                ;SKIP TO CLEAR
 4111  015134  032737  040000  002456  51$:    BIT     #DRVERR,T.CS      ;TEST IF DRIVE ERROR NOW
 4112  015142  001031                          BNE     5$                ;YES - SKIP TO CLEAR
 4113  015144                                  WAITMS  #1                ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HC
   (3) 015144  012700  000001                  MOV     #1,R0
   (3) 015150  104026                          EMT     C$WTM
 4114  015152  005301                          DEC     R1                ;DEC WAIT COUNTER
 4115  015154  001344                          BNE     50$               ;IF NOT DONE, LOOP
 4116  015156  012703  007771                  MOV     #MUNDEF,R3        ;MESSAGE FOR UNDEFINED STATE
 4117  015162                                  ERRHRD  10001.,,ERR1
   (3) 015162  104443                          TRAP    T$ERCODE
   (5) 015164  023421                          .WORD   10001
   (5) 015166  011344                          .WORD   ERR1
 4118  015170  000554                          BR      14$               ;EXIT
 4119  015172  005737  002540          11$:    TST     TEMP4             ;TEST IF SAVE REGISTERS
 4120  015176  001013                          BNE     5$                ;NO SKIP
 4121  015200  012701  000004                  MOV     #4,R1             ;SET SAVE COUNT
 4122  015204  012703  002456                  MOV     #L.MP+2,R3        ;SET ADDRESS OF FIRST SAVE
 4123  015210  014346                  8$:     MOV     -(R3),-(SP)       ;PUT REG ON STACK
 4124  015212  005301                          DEC     R1                ;DEC COUNT
 4125  015214  001375                          BNE     8$                ;LOOP UNTIL ALL SAVED
 4126  015216  012737  000003  002452          MOV     #GETSTAT,L.DA     ;SET FOR GET STATUS
 4127  015224  000403                          BR      6$                ;SKIP
 4128  015226  013737  002540  002452  5$:     MOV     TEMP4,L.DA        ;INSERT PRESET FOR STATUS
 4129  015234                          6$:
 4130  015234  005037  002420                  CLR     DONE              ;CLEAR INTERRUPT FLAG
 4131  015240  013737  002444  002446          MOV     RLDRV,L.CS        ;SET UP TO GET STATUS
 4132  015246  042737  002000  002446          BIC     #BIT10,L.CS       ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4133  015254  052737  000104  002446          BIS     #GTSTAT,L.CS
 4134  015262  013762  002452  000004          MOV     L.DA,RLDA(R2)     ;LOAD RL REGS
 4135  015270  013762  002446  000000          MOV     L.CS,RLCSR(R2)    ;LOAD CS REG
 4136  015276                                  WAITUS  #1                ;WAIT 100 US FOR INTERRUPT
   (3) 015276  012700  000001                  MOV     #1,R0
   (3) 015302  104027                          EMT     C$WTU
 4137  015304  005737  002420                  TST     DONE              ;CHECK IF INTERRUPT OCCURRED
 4138  015310  001476                          BEQ     1$                ;NO - SKIP
 4139  015312  013737  002464  002472  4$:     MOV     T.MP,T.STAT       ;STORE MP REGISTER
```

```
 4140  015320  042737  177770  002472          BIC     # C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
 4141  015326  032737  000010  002452          BIT     #DRSET,L.DA       ;TEST IF RESET WAS SPECIFIED
 4142  015334  001474                          BEQ     3$                ;NO - SKIP TO EXIT
 4143  015336  032737  040000  002416          BIT     #RELDWT,OPFLAG    ;TEST IF RELOAD WAIT FLAG SET
 4144  015344  001424                          BEQ     12$               ;NO - SKIP
 4145  015346  012701  001130                  MOV     #600.,R1          ;SET WAIT COUNT FOR 60 SECONDS
 4146  015352  032762  000001  000000  13$:    BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
 4147  015360  001016                          BNE     12$               ;YES - SKIP
 4148  015362                                  WAITMS  #1                ;CALL WAIT
   (3) 015362  012700  000001                  MOV     #1,R0
   (3) 015366  104026                          EMT     C$WTM
 4149  015370  005301                          DEC     R1                ;DEC COUNT
 4150  015372  001367                          BNE     13$               ;LOOP IF NOT 0
 4151  015374  004737  014762                  JSR     PC,GSTAT          ;GET DRIVE STATUS
 4152  015400  015526                          3$                        ;ERROR RETURN
 4153  015402  012703  010036                  MOV     #MRLFAL,R3        ;SET RESULT MESSAGE POINTER
 4154  015406                                  ERRHRD  10003.,,ERR1
   (3) 015406  104443                          TRAP    T$ERCODE
   (5) 015410  023423                          .WORD   10003
   (5) 015412  011344                          .WORD   ERR1
 4155  015414  000442                          BR      14$               ;GO TO EXIT
 4156  015416                          12$:    WAITUS  #10.              ;WAIT FOR 1MS
   (3) 015416  012700  000012                  MOV     #10.,R0
   (3) 015422  104027                          EMT     C$WTU
 4157  015424  004737  014762                  JSR     PC,GSTAT          ;GET DRIVE STATUS
 4158  015430  015526                          3$
 4159  015432  032737  100000  002456          BIT     #ANYERR,T.CS      ;TEST IF ANY ERROR
 4160  015440  001432                          BEQ     3$                ;NO - SKIP
 4161  015442  032737  001000  002464          BIT     #VCSTAT,T.MP      ;CHECK IF VOLUME CHECK RESET
 4162  015450  001403                          BEQ     7$                ;YES SKIP
 4163  015452  012703  005751                  MOV     #VCNRST,R3        ;SET REASON POINTER
 4164  015456  000416                          BR      2$                ;EXIT
 4165  015460  032737  040000  002456  7$:     BIT     #DRVERR,T.CS      ;CHECK IF DRIVE ERROR
 4166  015466  001404                          BEQ     9$                ;NO - SKIP
 4167  015470                                  ERRHRD  10004.,,ERR6
   (3) 015470  104443                          TRAP    T$ERCODE
   (5) 015472  023424                          .WORD   10004
   (5) 015474  011646                          .WORD   ERR6
 4168  015476  000411                          BR      14$               ;EXIT
 4169  015500  012703  005772          9$:     MOV     #UNXERR,R3        ;SET REASON POINTER
 4170  015504  000403                          BR      2$                ;EXIT
 4171  015506  004737  014562          1$:     JSR     PC,WAITIN         ;WAIT FOR INTERRUPT
 4172  015512  012603                          MOV     (SP)+,R3          ;STORE REASON POINTER FOR RETURN
 4173  015514                          2$:     ERRHRD  10002.,,ERR1
   (3) 015514  104443                          TRAP    T$ERCODE
   (5) 015516  023422                          .WORD   10002
   (5) 015520  011344                          .WORD   ERR1
 4174  015522  005037  002430          14$:    CLR     ERRSWI            ;CLEAR FOR ERROR RETURN
 4175  015526  005737  002540          3$:     TST     TEMP4             ;TEST IF REGISTERS WERE SAVED
 4176  015532  001007                          BNE     22$               ;NO - SKIP
 4177  015534  012703  002446                  MOV     #L.CS,R3          ;SET POINTER TO RESTORE
 4178  015540  012701  000004                  MOV     #4,R1             ;SET REGISTER COUNT
 4179  015544  012623                  20$:    MOV     (SP)+,(R3)+       ;RESTORE REG
 4180  015546  005301                          DEC     R1                ;DEC COUNT
 4181  015550  001375                          BNE     20$               ;LOOP UNTIL ALL ARE RESTORED
 4182  015552  162737  000002  002414  22$:    SUB     #2,SSINDX         ;REMOVE ENTRY FROM SUBROUT STACK
```

```
4183  015560  012601                        MOV     (SP)+,R1        ;RESTORE R1
4184  015562  012600                        MOV     (SP)+,R0
4185  015564  012603                        MOV     (SP)+,R3        ;RESTORE R3
4186  015566  012637  002540                MOV     (SP)+,TEMP4     ;RESTORE TEMP4
4187  015572  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
4188  015576  001403                        BEQ     99$             ;YES - SKIP
4189  015600  063716  002430                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
4190  015604  000207                        RTS     PC
4191  015606  017616  000000        99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
4192  015612  000207                        RTS     PC
4193
4194
4196                                ;       SEEK ROUTINE
4197  015614  012737  177777  002532 XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
4198  015622  000402                        BR      XSEEK1
4199  015624  005037  002532        XSEEK:  CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
4200  015630  010346                XSEEK1: MOV     R3,-(SP)        ;STORE R3
4201  015632  013703  002414                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
4202  015636  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
4203  015640  016663  000002  002250        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
4204  015646  162763  000004  002250        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
4205  015654  010337  002414                MOV     R3,SSINDX       ;STORE IT BACK
4206  015660  010046                        MOV     R0,-(SP)
4207  015662  010146                        MOV     R1,-(SP)
4208  015664  010546                        MOV     R5,-(SP)        ;STORE REG
4209  015666  012737  000002  002430        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
4210  015674  005037  002510                CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
4211                                                                ;    PAST GUARD BAND)
4212  015700  004737  020652                JSR     PC,GETPOS       ;GET PRESENT POSITION
4213  015704  016264                        #65$
4214  015706  013737  002516  002512        MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
4215  015714  023727  002514  000377        CMP     NEWCYL,#255.    ;TEST IF NEW IS GREATER THAN 255
4216  015722  003412                        BLE     3$              ;NO - SKIP
4217  015724  162737  000377  002514        SUB     #255.,NEWCYL    ;ELSE SUBTRACT 255.
4218  015732  013737  002514  002510        MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
4219  015740  012737  000377  002514        MOV     #255.,NEWCYL    ;SET NEWCYL AS 255.
4220  015746  000412                        BR      6$              ;SKIP
4221  015750  005737  002514        3$:     TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
4222  015754  100007                        BPL     6$              ;NO - SKIP
4223  015756  005437  002514                NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
4224  015762  013737  002514  002510        MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
4225  015770  005037  002514                CLR     NEWCYL          ;AND SET NEWCYL TO 0
4226  015774  013705  002516        6$:     MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
4227  016000  163705  002514                SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
4228  016004  100005                        BPL     13$             ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
4229  016006  012737  000001  002522        MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
4230  016014  005405                        NEG     R5              ;MAKE DIFFERENCE POSITIVE
4231  016016  000402                        BR      14$             ;SKIP
4232  016020  005037  002522        13$:    CLR     DESSGN          ;SET SIGN FOR REVERSE
4233  016024  010537  002520        14$:    MOV     R5,DESDIF       ;STORE DIFFERENCE
4234  016030  005737  002510                TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
4235  016034  001412                        BEQ     18$             ;NO - SKIP
4236  016036  023727  002514  000377        CMP     NEWCYL,#255.    ;CHECK IF NEW CYL IS 255.
4237  016044  001003                        BNE     17$             ;NO - SKIP
4238  016046  012737  000001  002522        MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
4239                                                                ;(INNER GUARD BAND)
```

```
 4240  016054  063737  002510  002520  17$:    ADD     DIFAUG,DESDIF   ;ADD ANY AUGMENT TO DIFFERENCE
 4241  016062                          18$:
 4242  016062  012705  002446                   MOV     #L.CS,R5        ;GET L REG ADDRESS
 4243  016066  012715  000106                   MOV     #SEEK,(R5)      ;SET FOR SEEK
 4244  016072  053715  002444                   BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
 4245  016076  042725  002000                   BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
 4246  016102  005025                          CLR     (R5)+           ;CLEAR BUS ADDRESS
 4247  016104  013715  002520                   MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
 4248  016110  012700  000007                   MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
 4249  016114  006315                  21$:    ASL     (R5)
 4250  016116  005300                          DEC     R0
 4251  016120  001375                          BNE     21$             ;LOOP UNTIL ALIGNED
 4252  016122  005737  002522                   TST     DESSGN          ;TEST SIGN
 4253  016126  001402                          BEQ     23$             ;SKIP IF 0
 4254  016130  052715  000004                   BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
 4255  016134  005737  002524          23$:    TST     DESHD           ;TEST IF HEAD 0
 4256  016140  001402                          BEQ     25$             ;YES - SKIP
 4257  016142  052715  000020                   BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
 4258  016146  052725  000001          25$:    BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
 4259  016152  004737  016664                   JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
 4260  016156  016264                          #65$
 4261  016160  005037  002420                   CLR     DONE            ;CLEAR INTERRUPT FLAG
 4262  016164  005737  002532                   TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
 4263  016170  001035                          BNE     65$             ;YES - SKIP, DO NOT START SEEK
 4264  016172  014562  000004                   MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
 4265  016176  014562  000002                   MOV     -(R5),RLBA(R2)
 4266  016202  014562  000000                   MOV     -(R5),RLCS(R2)
 4267  016206                          30$:    WAITUS  #10.
  (3)  016206  012700  000012                   MOV     #10.,R0
  (3)  016212  104027                          EMT     C$WTU
 4268  016214  005737  002420                   TST     DONE            ;TEST IF INTERRUPT DONE
 4269  016220  001011                          BNE     32$             ;YES - SKIP
 4270  016222  004737  014562                   JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
 4271  016226  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 4272  016230                                  ERRHRD  10005.,,ERR1
  (3)  016230  104443                          TRAP    T$ERCODE
  (5)  016232  023425                          .WORD   10005
  (5)  016234  011344                          .WORD   ERR1
 4273  016236  005037  002430                   CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4274  016242  000410                          BR      65$
 4275  016244  005737  002456          32$:    TST     T.CS            ;TEST IF ANY ERROR
 4276  016250  100005                          BPL     65$             ;NO - SKIP
 4277  016252                                  ERRHRD  10006.,,ERR6
  (3)  016252  104443                          TRAP    T$ERCODE
  (5)  016254  023426                          .WORD   10006
  (5)  016256  011646                          .WORD   ERR6
 4278  016260  005037  002430                   CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4279  016264  162737  000002  002414  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4280  016272  012605                          MOV     (SP)+,R5        ;RESTORE REGISTER
 4281  016274  012601                          MOV     (SP)+,R1
 4282  016276  012600                          MOV     (SP)+,R0
 4283  016300  012603                          MOV     (SP)+,R3        ;RESTORE R3
 4284  016302  005737  002430                   TST     ERRSWI          ;TEST IF ERROR RETURN
 4285  016306  001403                          BEQ     99$             ;YES - SKIP
 4286  016310  063716  002430                   ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4287  016314  000207                          RTS     PC
```

```
 4288  016316  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4289  016322  000207                          RTS     PC
 4290
 4348
 4350                                  ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
 4351                                  ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
 4352  016324  010346                  POSHDS: MOV     R3,-(SP)        ;SAVE REGS
 4353  016326  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4354  016332  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4355  016334  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4356  016342  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4357  016350  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4358  016354  010346                          MOV     R3,-(SP)
 4359  016356  010446                          MOV     R4,-(SP)
 4360  016360  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4361  016366  004737  020652                  JSR     PC,GETPOS       ;GET CURRENT POSITION
 4362  016372  016626                          #PH65$
 4363  016374  012704  000012                  MOV     #10.,R4         ;SET RETRY COUNT
 4364  016400                          BGNSEG
  (3)  016400  104004                          EMT     C$BSEG
 4365  016402                          1$:     INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  016402  104020                          EMT     C$INLP
 4366  016404                                  BNCOMPLETE      5$      ;NO - SKIP
  (2)  016404  103012                          BCC     5$
 4367  016406  004737  020652                  JSR     PC,GETPOS       ;ELSE GET POSITION
 4368  016412  016624                          #60$
 4369  016414  023737  002516  002514          CMP     CURCYL,NEWCYL   ;CHECK IF AT INTENDED POSITION
 4370  016422  001017                          BNE     8$              ;NO - SKIP
 4371  016424  004737  017210                  JSR     PC,ONSWAP       ;SWAP OLDCYL AND NEWCYL
 4372  016430  000414                          BR      8$              ;SKIP
 4373  016432  013737  002516  002512  5$:     MOV     CURCYL,OLDCYL   ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
 4374  016440  023705  002516                  CMP     CURCYL,R5       ;CHECK IF HDS AT FINAL POSITION
 4375  016444  001467                          BEQ     60$             ;YES - GO TO EXIT
 4376  016446  003003                          BGT     7$              ;IF CURCYL > FINAL POSITION - SKIP
 4377  016450  005237  002514                  INC     NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
 4378  016454  000402                          BR      8$              ;SKIP
 4379  016456  005337  002514          7$:     DEC     NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
 4380  016462  004737  015624          8$:     JSR     PC,XSEEK        ;DO SEEK
 4381  016466  016624                          #60$
 4382  016470  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT 300 MS
 4383  016474  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
 4384  016500  016624                          #60$
 4385  016502  005737  002456                  TST     T.CS            ;TEST IF ANY ERROR
 4386  016506  100006                          BPL     10$             ;NO - SKIP
 4387  016510                                  ERRHRD  10008.,,ERR6
  (3)  016510  104443                          TRAP    T$ERCODE
  (5)  016512  023430                          .WORD   10008
  (5)  016514  011646                          .WORD   ERR6
 4388  016516  005037  002430                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4389  016522  000440                          BR      60$
 4390  016524  004737  020652          10$:    JSR     PC,GETPOS       ;GET POSITION
 4391  016530  016624                          #60$
 4392  016532  023737  002516  002514          CMP     CURCYL,NEWCYL   ;CHECK IF ARRIVED AT DESIRED PLACE
 4393  016540  001003                          BNE     15$             ;NO - SKIP
 4394  016542  012704  000012          14$:    MOV     #10.,R4         ;ELSE INIT RETRY COUNT
 4395  016546  000715                          BR      1$              ;GO DO NEXT SEEK
```

```
 4396  016550  005737  002522          15$:    TST     DESSGN          ;TEST IF GOING IN
 4397  016554  001016                  BNE     17$             ;YES - SKIP
 4398  016556  023737  002516  002514          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS DID NOT MOVE IN
 4399  016564  003366                  BGT     14$             ;YES - SKIP
 4400  016566  005304                  16$:    DEC     R4              ;DEC RETRY COUNT
 4401  016570  001334                  BNE     8$              ;DO ANOTHER SEEK IF NOT 0
 4402  016572  012703  006762                  MOV     #HDMOVF,R3      ;ELSE SET RESULT MESSAGE POINTER
 4403  016576                          ERRHRD  10009.,,ERR1
   (3) 016576  104443                  TRAP    T$ERCODE
   (5) 016600  023431                  .WORD   10009
   (5) 016602  011344                  .WORD   ERR1
 4404  016604  005037  002430                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4405  016610  000405                  BR      60$
 4406  016612  023737  002516  002514  17$:    CMP     CURCYL,NEWCYL   ;HDS SHOULD MOVE OUT, CHK THEY DID
 4407  016620  002750                  BLT     14$             ;YES - SKIP
 4408  016622  000761                  BR      16$             ;ELSE GO DEC AND RETRY
 4409  016624                          20$:
 4410  016624                          60$:
 4411  016624                          ENDSEG
   (3) 016624                          10000$:
   (3) 016624  104005                  EMT     C$ESEG
 4412  016626  162737  000002  002414  PH65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4413  016634  012604                  MOV     (SP)+,R4        ;RESTORE REGISTERS
 4414  016636  012600                  MOV     (SP)+,R0
 4415  016640  012603                  MOV     (SP)+,R3
 4416  016642  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4417  016646  001403                  BEQ     99$             ;YES - SKIP
 4418  016650  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4419  016654  000207                  RTS     PC
 4420  016656  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4421  016662  000207                  RTS     PC
 4422
 4424                                  ;       DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT
 4425                                  ;       500MS FOR READY TO SET.
 4426  016664  010346                  RDYCHK: MOV     R3,-(SP)        ;STORE REGS
 4427  016666  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4428  016672  005723                  TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4429  016674  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4430  016702  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4431  016710  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4432  016714  010046                  MOV     R0,-(SP)
 4433  016716  010146                  MOV     R1,-(SP)
 4434  016720  010446                  MOV     R4,-(SP)
 4435  016722  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4436  016730  012701  011610                  MOV     #5000.,R1       ;SET WAIT COUNT
 4437  016734  004737  014762          1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 4438  016740  017060                  4$
 4439  016742  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 4440  016750  001045                  BNE     5$              ;YES - EXIT
 4441  016752                          WAITUS  #1
   (3) 016752  012700  000001                  MOV     #1,R0
   (3) 016756  104027                  EMT     C$WTU
 4442  016760  005301                  DEC     R1              ;DEC WAIT COUNT
 4443  016762  001364                  BNE     1$              ;LOOP IF NOT 0
 4444  016764  012703  007333                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
 4445  016770  012704  010261                  MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
```

```
 4446  016774                                        ERRHRD  10010.,,ERR5
  (3)  016774  104443                                TRAP    T$ERCODE
  (5)  016776  023432                                .WORD   10010
  (5)  017000  011576                                .WORD   ERR5
 4447  017002  012701  000062                        MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 4448  017006  004737  014762          2$:           JSR     PC,GSTAT        ;GET DRIVE STATUS
 4449  017012  017060                                4$
 4450  017014  032737  000001  002456                BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 4451  017022  001005                                BNE     3$              ;YES - SKIP
 4452  017024                                        WAITMS  #1              ;WAIT FOR 100MS
  (3)  017024  012700  000001                        MOV     #1,R0
  (3)  017030  104026                                EMT     C$WTM
 4453  017032  005301                                DEC     R1              ;DEC WAIT COUNTER
 4454  017034  001364                                BNE     2$              ;LOOP UNTIL TIME DONE
 4455  017036  032737  100000  002456  3$:           BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
 4456  017044  001405                                BEQ     4$              ;NO - SKIP
 4457  017046                                        ERRHRD  10011.,,ERR6    ;REPORT ALL ERRORS
  (3)  017046  104443                                TRAP    T$ERCODE
  (5)  017050  023433                                .WORD   10011
  (5)  017052  011646                                .WORD   ERR6
 4458  017054  005337  002650                        DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
 4459  017060  005037  002430          4$:           CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4460  017064  162737  000002  002414  5$:           SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4461  017072  012604                                MOV     (SP)+,R4        ;RESTORE REGS
 4462  017074  012601                                MOV     (SP)+,R1
 4463  017076  012600                                MOV     (SP)+,R0
 4464  017100  012603                                MOV     (SP)+,R3
 4465  017102  005737  002430                        TST     ERRSWI          ;TEST IF ERROR RETURN
 4466  017106  001403                                BEQ     99$             ;YES - SKIP
 4467  017110  063716  002430                        ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4468  017114  000207                                RTS     PC
 4469  017116  017616  000000          99$:          MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4470  017122  000207                                RTS     PC
 4471
 4473                                    ;           CHOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
 4474                                    ;           SELECTED BY SOFTWARE PARAMETER.
 4475  017124  005037  002524          CHOSHD:       CLR     DESHD           ;CLEAR TO HEAD 0
 4476  017130  032737  010000  013162                BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
 4477  017136  001403                                BEQ     1$              ;NO - SKIP
 4478  017140  013737  013170  002524                MOV     HEADW,DESHD     ;INSERT SPECIFIED HEAD
 4479  017146  000207                          1$:   RTS     PC
 4480
 4481                                    ;           SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
 4482                                    ;           UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
 4483  017150  032737  010000  013162  SWAPHD:       BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
 4484  017156  001011                                BNE     2$              ;YES - TAKE ABORT EXIT
 4485  017160  005737  002524                        TST     DESHD           ;TEST IF HEAD ONE USED
 4486  017164  001006                                BNE     2$              ;YES - TAKE ABORT EXIT
 4487  017166  012737  000001  002524                MOV     #1,DESHD        ;ELSE SET FOR HEAD ONE
 4488  017174  062716  000002                        ADD     #2,(SP)         ;BUMP PAST ABORT RETURN
 4489  017200  000207                                RTS     PC              ;RETURN
 4490  017202  017616  000000          2$:           MOV     @(SP),(SP)      ;GET ABORT DESTINATION
 4491  017206  000207                    3$:         RTS     PC
 4492
 4493                                    ;           SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
 4494  017210  010046                    ONSWAP:     MOV     R0,-(SP)        ;STORE R0
```

```
 4495  017212  013700  002512                    MOV     OLDCYL,R0       ;MOVE OLD TO R0
 4496  017216  013737  002514  002512            MOV     NEWCYL,OLDCYL   ;MOVE NEW TO OLD
 4497  017224  010037  002514                    MOV     R0,NEWCYL       ;PUT OLD IN NEW
 4498  017230  012600                            MOV     (SP)+,R0        ;RESTORE R0
 4499  017232  000207                            RTS     PC
 4500
 4501                                  ;         BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
 4502                                  ;         FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
 4503                                  ;         FORCE FILES TO LOOK LIKE ALL SECTORS OK.
 4504  017234  005737  002664          CKBSVD:  TST     BSFVAL          ;TEST IF BAD SECTORS STORED
 4505  017240  001051                            BNE     5$              ;YES - EXIT
 4506  017242                                    PRINTF  #FMT9,#BSNSTR   ;REPORT
  (8)  017242  012746  007232                    MOV     #BSNSTR,-(SP)
  (7)  017246  012746  010633                    MOV     #FMT9,-(SP)
  (6)  017252  012746  000002                    MOV     #2,-(SP)
  (3)  017256  010600                            MOV     SP,R0
  (4)  017260  104017                            EMT     C$PNTF
  (4)  017262  062706  000006                    ADD     #6,SP
 4507  017266                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  017266  005046                            CLR     -(SP)
 (11)  017270  153716  002445                    BISB    RLDRV+1,(SP)
 (10)  017274  012746  005423                    MOV     #DRVNAM,-(SP)
  (9)  017300  013746  002440                    MOV     RLBAS,-(SP)
  (8)  017304  012746  005412                    MOV     #BASADD,-(SP)
  (7)  017310  012746  010447                    MOV     #FMT5,-(SP)
  (6)  017314  012746  000005                    MOV     #5,-(SP)
  (3)  017320  010600                            MOV     SP,R0
  (4)  017322  104017                            EMT     C$PNTF
  (4)  017324  062706  000014                    ADD     #14,SP
 4508  017330                                    PRINTF  #FMT3
  (7)  017330  012746  010433                    MOV     #FMT3,-(SP)
  (6)  017334  012746  000001                    MOV     #1,-(SP)
  (3)  017340  010600                            MOV     SP,R0
  (4)  017342  104017                            EMT     C$PNTF
  (4)  017344  062706  000004                    ADD     #4,SP
 4509  017350  012737  177777  002666            MOV     #-1,SBSFIL      ;FORCE FILES TO NO ENTRIES
 4510  017356  012737  177777  003062            MOV     #-1,FBSFIL
 4511  017364  000207                    5$:     RTS     PC
 4512
 4514                                  ;         READ HEADERS ROUTINE.
 4515  017366  012737  000001  002540  XRDHDC:  MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
 4516  017374  000402                            BR      XRDHDG          ;GO DO IT
 4517  017376  005037  002540          XRDHD:   CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
 4518  017402  010346                  XRDHDG:  MOV     R3,-(SP)        ;STORE REGISTERS
 4519  017404  013703  002414                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4520  017410  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4521  017412  016663  000002  002250            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4522  017420  162763  000004  002250            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4523  017426  010337  002414                    MOV     R3,SSINDX       ;STORE !T BACK
 4524  017432  010046                            MOV     R0,-(SP)
 4525  017434  010146                            MOV     R1,-(SP)
 4526  017436  010446                            MOV     R4,-(SP)
 4527  017440  012737  000002  002430            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4528  017446  005737  002540                    TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
 4529  017452  001007                            BNE     2$              ;NO - SKIP
 4530  017454  012703  002456                    MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
```

```
 4531  017460  012701  000004                 MOV     #4,R1                ;SET COUNT
 4532  017464  014346                  1$:    MOV     -(R3),-(SP)          ;SAVE REGISTER
 4533  017466  005301                         DEC     R1                   ;DEC COUNT
 4534  017470  001375                         BNE     1$                   ;LOOP UNTIL ALL ARE SAVED
 4535  017472  004737  016664           2$:    JSR     PC,RDYCHK            ;CHECK DRIVE READY
 4536  017476  017732                         #65$
 4537  017500  005037  002420                 CLR     DONE                 ;CLEAR INTERRUPT FLAG
 4538  017504  012701  002446                 MOV     #L.CS,R1             ;GET ADDRESS OF LOAD REGS
 4539  017510  013711  002444                 MOV     RLDRV,(R1)           ;LOAD DRIVE NUMBER
 4540  017514  042711  002000                 BIC     #BIT10,(R1)          ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4541  017520  052721  000110                 BIS     #RDHEAD,(R1)+        ;INSERT COMMAND
 4542  017524  005021                         CLR     (R1)+                ;CLEAR BA
 4543  017526  005021                         CLR     (R1)+                ;CLEAR DA
 4544  017530  014162  000004                 MOV     -(R1),RLDA(R2)       ;LOAD RL11 REGS
 4545  017534  014162  000002                 MOV     -(R1),RLBA(R2)
 4546  017540  014162  000000                 MOV     -(R1),RLCSR(R2)
 4547  017544                          3$:    WAITUS  #10.                 ;WAIT 1MS FOR INTERRUPT
  (3)  017544  012700  000012                 MOV     #10.,R0
  (3)  017550  104027                         EMT     C$WTU
 4548  017552  005737  002420                 TST     DONE                 ;TEST IN INTERRUPT FLAG SET
 4549  017556  001455                         BEQ     14$                  ;NO - SKIP
 4550  017560  032737  000001  002456   5$:    BIT     #DRDYMSK,T.CS        ;TEST IF DRIVE READY
 4551  017566  001033                         BNE     10$                  ;YES - SKIP
 4552  017570  012703  007333                 MOV     #MDRDY,R3            ;SET NO READY MESSAGE
 4553  017574  012704  010300                 MOV     #CAFDT,R4            ;CONDITION OF AFTER DATA XFER
 4554  017600                                  ERRHRD  10017.,,ERR5
  (3)  017600  104443                         TRAP    T$ERCODE
  (5)  017602  023441                         .WORD   10017
  (5)  017604  011576                         .WORD   ERR5
 4555  017606  012701  000062                 MOV     #50.,R1              ;SET WAIT COUNT FOR 5 SECONDS
 4556  017612  004737  014762           4$:    JSR     PC,GSTAT             ;GET STATUS
 4557  017616  017726                         60$
 4558  017620  032737  000001  002456         BIT     #DRDYMSK,T.CS        ;TEST IF DRIVE HAS COME READY
 4559  017626  001403                         BEQ     11$                  ;NO - SKIP
 4560  017630  005037  002430                 CLR     ERRSWI               ;CLEAR ERROR SWITCH
 4561  017634  000410                         BR      10$                  ;SKIP
 4562  017636  005301                  11$:   DEC     R1                   ;DEC WAIT COUNT
 4563  017640  001364                         BNE     4$                   ;LOOP UNTIL TIME DONE
 4564  017642  012704  010312                 MOV     #C5SEC,R4            ;SET CONDITION AFTER 5 SECONDS
 4565  017646                                  ERRHRD  10014.,,ERR5
  (3)  017646  104443                         TRAP    T$ERCODE
  (5)  017650  023436                         .WORD   10014
  (5)  017652  011576                         .WORD   ERR5
 4566  017654  000424                         BR      60$                  ;EXIT
 4567  017656  005737  002456           10$:   TST     T.CS                 ;CHECK FOR ANY ERRORS
 4568  017662  100004                         BPL     12$                  ;NO - SKIP
 4569  017664                                  ERRHRD  10016.,,ERR6         ;REPORT ALL ERRORS
  (3)  017664  104443                         TRAP    T$ERCODE
  (5)  017666  023440                         .WORD   10016
  (5)  017670  011646                         .WORD   ERR6
 4570  017672  000415                         BR      60$
 4571  017674  012701  002466           12$:   MOV     #HDWRD2,R1           ;GET POINTER
 4572  017700  016221  000006                 MOV     RLMP(R2),(R1)+       ;STORE LAST TWO HEADER WORDS
 4573  017704  016221  000006                 MOV     RLMP(R2),(R1)+
 4574  017710  000410                         BR      65$                  ;EXIT
 4575  017712  004737  014562           14$:   JSR     PC,WAITIN            ;WAIT FOR INTERRUPT
```

```
 4576  017716  012603                        MOV     (SP)+,R3        ;GET RESULTS
 4577  017720                                ERRHRD  10015.,,ERR1    ;REPORT
  (3)  017720  104443                        TRAP    T$ERCODE
  (5)  017722  023437                        .WORD   10015
  (5)  017724  011344                        .WORD   ERR1
 4578  017726  005037  002430          60$:  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4579  017732  005737  002540          65$:  TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 4580  017736  001007                        BNE     22$             ;NO - SKIP
 4581  017740  012703  002446                MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
 4582  017744  012701  000004                MOV     #4,R1           ;SET COUNT
 4583  017750  012623                  20$:  MOV     (SP)+,(R3)+     ;RESTORE REGISTER
 4584  017752  005301                        DEC     R1              ;DEC COUNT
 4585  017754  001375                        BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
 4586  017756  162737  000002  002414  22$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4587  017764  012604                        MOV     (SP)+,R4        ;RESTORE REGS
 4588  017766  012601                        MOV     (SP)+,R1
 4589  017770  012600                        MOV     (SP)+,R0
 4590  017772  012603                        MOV     (SP)+,R3
 4591  017774  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
 4592  020000  001403                        BEQ     99$             ;YES - SKIP
 4593  020002  063716  002430                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4594  020006  000207                        RTS     PC
 4595  020010  017616  000000          99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4596  020014  000207                        RTS     PC
 4597
 4599                                  ;     VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
 4600                                  ;     SEQUENCE.
 4601  020016  010346                  VERHDR: MOV     R3,-(SP)        ;STORE REGS
 4602  020020  013703  002414                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4603  020024  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4604  020026  016663  000002  002250        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4605  020034  162763  000004  002250        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4606  020042  010337  002414                MOV     R3,SSINDX       ;STORE IT BACK
 4607  020046  010046                        MOV     R0,-(SP)
 4608  020050  010146                        MOV     R1,-(SP)
 4609  020052  010446                        MOV     R4,-(SP)
 4610  020054  010546                        MOV     R5,-(SP)
 4611  020056  012737  000002  002430        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4612  020064  052737  000002  002416        BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
 4613  020072  005037  002426                CLR     MORECE          ;CLEAR MORE ERRORS FLAG
 4614  020076  012704  003256                MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
 4615  020102  012705  002530                MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
 4616  020106  005003                        CLR     R3              ;CLEAR FOR WORD COUNTER
 4617  020110  011415                        MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
 4618  020112  011401                        MOV     (R4),R1         ;PUT WORD IN REG 1
 4619  020114  042701  100177                BIC     # CHDCYL,R1     ;CLEAR ALL BUT CYLINDER
 4620  020120  012700  000007                MOV     #7,R0           ;SET SHIFT COUNT
 4621  020124  006201                  3$:   ASR     R1              ;SHIFT
 4622  020126  005300                        DEC     R0              ;DEC
 4623  020130  001375                        BNE     3$              ;LOOP
 4624  020132  020137  002514                CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
 4625  020136  001406                        BEQ     4$              ;YES - SKIP
 4626  020140                                ERRHRD  10018.,,ERR10   ;REPORT ERROR
  (3)  020140  104443                        TRAP    T$ERCODE
  (5)  020142  023442                        .WORD   10018
  (5)  020144  012734                        .WORD   ERR10
```

```
 4627  020146  005037  002430                CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4628  020152  000454                        BR      65$
 4629  020154  012701  000050          4$:   MOV     #40.,R1         ;SET HEADER COUNT
 4630  020160  042715  100100                BIC     #BIT15!HDHSEL,(R5)      ;CLEAR HEAD SELECT AND 0 BIT
 4631  020164  005737  002524                TST     DESHD           ;ARE WE USING HD 0?
 4632  020170  001402                        BEQ     5$              ;YES - SKIP
 4633  020172  052715  000100                BIS     #HDHSEL,(R5)    ;INSERT HEAD BIT
 4634  020176  005065  000002          5$:   CLR     2(R5)           ;CLEAR 2ND WORD OF WORK AREA
 4635  020202  021524                6$:   CMP     (R5),(R4)+      ;TEST FIRST WORD OK
 4636  020204  001407                        BEQ     8$              ;YES - SKIP
 4637  020206  005744                        TST     -(R4)           ;ELSE SET POINTER FOR ERROR
 4638  020210                                ERRHRD  10018.,,ERR10   ;REPORT
  (3)  020210  104443                        TRAP    T$ERCODE
  (5)  020212  023442                        .WORD   10018
  (5)  020214  012734                        .WORD   ERR10
 4639  020216  005037  002430                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4640  020222  005724                        TST     (R4)+           ;RESET POINTER
 4641  020224  005203                8$:   INC     R3              ;BUMP WORD COUNTER
 4642  020226  005724                        TST     (R4)+           ;TEST 2ND WORD IS 0
 4643  020230  001407                        BEQ     12$             ;YES - SKIP
 4644  020232  022544                        CMP     (R5)+,-(R4)     ;ADJUST POINTERS FOR REPORT
 4645  020234                                ERRHRD  10018.,,ERR10   ;REPORT
  (3)  020234  104443                        TRAP    T$ERCODE
  (5)  020236  023442                        .WORD   10018
  (5)  020240  012734                        .WORD   ERR10
 4646  020242  005037  002430                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4647  020246  024524                        CMP     -(R5),(R4)+     ;RESET POINTERS
 4648  020250  005724                12$:   TST     (R4)+           ;BUMP PAST ECC WORD
 4649  020252  005203                        INC     R3              ;BUMP WORD COUNTER
 4650  020254  005215                        INC     (R5)            ;BUMP SECTOR OF EXPECTED HEADER
 4651  020256  011500                        MOV     (R5),R0         ;MOVE EXPECTED HDR TO R0
 4652  020260  042700  177700                BIC     # CHDSEC,R0     ;CLEAR ALL BUT SECTOR
 4653  020264  022700  000050                CMP     #40.,R0         ;TEST IF AT SECTOR 40
 4654  020270  001002                        BNE     15$             ;NO - SKIP
 4655  020272  042715  000077                BIC     #HDSEC,(R5)     ;CLEAR SECTOR TO 0
 4656  020276  005203                15$:   INC     R3              ;BUMP HDR WORD COUNTER
 4657  020300  005301                        DEC     R1              ;DEC HEADER COUNT
 4658  020302  001337                        BNE     6$              ;LOOP IF NOT YET DONE
 4659  020304  162737  000002  002414  65$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4660  020312  012605                        MOV     (SP)+,R5        ;RESTORE REGISTERS
 4661  020314  012604                        MOV     (SP)+,R4
 4662  020316  012601                        MOV     (SP)+,R1
 4663  020320  012600                        MOV     (SP)+,R0
 4664  020322  012603                        MOV     (SP)+,R3
 4665  020324  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
 4666  020330  001403                        BEQ     99$             ;YES - SKIP
 4667  020332  063716  002430                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4668  020336  000207                        RTS     PC
 4669  020340  017616  000000          99$:   MOV     @(SP),(SP)      ;SET ERPOR RETURN ADDRESS
 4670  020344  000207                        RTS     PC
 4671
 4673                                  ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
 4674  020346  013705  002464          POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
 4675  020352  000402                        BR      POSHDO          ;SKIP
 4676  020354  013705  002464          POSHSB: MOV     T.MP,R5 ;START FOR POSITION HD BIT IN MP
 4677  020360  010146                  POSHDO: MOV     R1,-(SP)        ;STORE R1
```

```
 4678  020362  042705  177677                  BIC     # CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
 4679  020366  012701  000006                  MOV     #6,R1           ;SET SHIFT COUNT
 4680  020372  006205                  1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
 4681  020374  005301                          DEC     R1
 4682  020376  001375                          BNE     1$
 4683  020400  012601                          MOV     (SP)+,R1        ;RESTORE R1
 4684  020402  000207                          RTS     PC              ;RETURN
 4685
 4686                                  ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
 4687                                  ;       FROM THE CALLING ROUTINE IN R1.
 4688  020404  010346                  RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
 4689  020406  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4690  020412  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4691  020414  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4692  020422  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4693  020430  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4694  020434  010046                          MOV     R0,-(SP)
 4695  020436  010146                          MOV     R1,-(SP)
 4696  020440  010446                          MOV     R4,-(SP)
 4697  020442  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4698  020450  004737  014762          5$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 4699  020454  020606                          10$
 4700  020456  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;CHECK IF READY
 4701  020464  001052                          BNE     9$              ;YES - SKIP
 4702  020466  005301                          DEC     R1              ;DEC WAIT COUNT
 4703  020470  001404                          BEQ     7$              ;SKIP IF 0
 4704  020472                                  WAITUS  #1
  (3)  020472  012700  000001                  MOV     #1,R0
  (3)  020476  104027                          EMT     C$WTU
 4705  020500  000763                          BR      5$
 4706  020502  012703  007333          7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 4707  020506                                  ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
  (3)  020506  104443                          TRAP    T$ERCODE
  (5)  020510  023444                          .WORD   10020
  (5)  020512  011460                          .WORD   ERR3
 4708  020514  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 4709  020520  004737  014762          6$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 4710  020524  020606                          10$
 4711  020526  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 4712  020534  001013                          BNE     8$              ;YES - SKIP
 4713  020536                                  WAITMS  #1              ;WAIT 100 MS
  (3)  020536  012700  000001                  MOV     #1,R0
  (3)  020542  104026                          EMT     C$WTM
 4714  020544  005301                          DEC     R1              ;DEC WAIT COUNT
 4715  020546  001364                          BNE     6$              ;LOOP UNTIL TIME DONE
 4716  020550  012704  010312                  MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
 4717  020554                                  ERRHRD  10021.,,ERR5
  (3)  020554  104443                          TRAP    T$ERCODE
  (5)  020556  023445                          .WORD   10021
  (5)  020560  011576                          .WORD   ERR5
 4718  020562  000407                          BR      11$             ;EXIT
 4719  020564  032737  100000  002456  8$:     BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
 4720  020572  001405                          BEQ     10$             ;NO - SKIP
 4721  020574                                  ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
  (3)  020574  104443                          TRAP    T$ERCODE
  (5)  020576  023446                          .WORD   10022
```

```
  (5)  020600  011646                          .WORD   ERR6
 4722  020602  005337  002650          11$:    DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
 4723  020606  005037  002430          10$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4724  020612  162737  000002  002414  9$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4725  020620  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
 4726  020622  012601                          MOV     (SP)+,R1
 4727  020624  012600                          MOV     (SP)+,R0
 4728  020626  012603                          MOV     (SP)+,R3        ;RESTORE R3
 4729  020630  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4730  020634  001403                          BEQ     99$             ;YES - SKIP
 4731  020636  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4732  020642  000207                          RTS     PC
 4733  020644  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4734  020650  000207                          RTS     PC
 4735
 4736                                  ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
 4737                                  ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
 4738                                  ;       NUMBER IN CURCYL.
 4739  020652  010346                  GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
 4740  020654  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4741  020660  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4742  020662  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4743  020670  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4744  020676  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4745  020702  010046                          MOV     R0,-(SP)
 4746  020704  010546                          MOV     R5,-(SP)
 4747  020706  004737  017376                  JSR     PC,XRDHD        ;DO READ HEADER
 4748  020712  020742                          #65$
 4749  020714  013703  002464                  MOV     HDWRD1,R3       ;GET HEADER WORD
 4750  020720  042703  100177                  BIC     # CHDCYL,R3     ;CLEAR ALL BUT CYLINDER
 4751  020724  012705  000007                  MOV     #7,R5           ;SET SHIFT COUNT
 4752  020730  006203                  4$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
 4753  020732  005305                          DEC     R5
 4754  020734  001375                          BNE     4$
 4755  020736  010337  002516                  MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
 4756  020742  162737  000002  002414  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4757  020750  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
 4758  020752  012600                          MOV     (SP)+,R0
 4759  020754  012603                          MOV     (SP)+,R3
 4760  020756  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4761  020762  001403                          BEQ     99$             ;YES - SKIP
 4762  020764  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4763  020770  000207                          RTS     PC
 4764  020772  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4765  020776  000207                          RTS     PC
 4766
 4768                                  ;       VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
 4769                                  ;       CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
 4770  021000  010346                  VERPOS: MOV     R3,-(SP)        ;STORE P3
 4771  021002  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4772  021006  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4773  021010  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4774  021016  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4775  021024  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4776
 4777  021030  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
```

```
 4778  021036  004737  020652                    JSR     PC,GETPOS       ;GET POSITION
 4779  021042  021066                            #65$
 4780  021044  023737  002514  002516            CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
 4781  021052  001405                            BEQ     1$              ;YES - SKIP
 4782  021054                                    ERRHRD  10022.,,ERR8
   (3) 021054  104443                            TRAP    T$ERCODE
   (5) 021056  023446                            .WORD   10022
   (5) 021060  012574                            .WORD   ERR8
 4783  021062  005037  002430                    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4784  021066                        1$:
 4785  021066  162737  000002  002414  65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4786  021074  012603                            MOV     (SP)+,R3        ;RESTORE R3
 4787  021076  005737  002430                    TST     ERRSWI          ;TEST IF ERROR RETURN
 4788  021102  001403                            BEQ     99$             ;YES - SKIP
 4789  021104  063716  002430                    ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4790  021110  000207                            RTS     PC
 4791  021112  017616  000000          99$:      MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4792  021116  000207                            RTS     PC
 4793
 4795                                    ;       READ ALL HEADERS ROUTINE.  40 HEADERS ARE READ AND STORED
 4796                                    ;       IN IBUFF.
 4797  021120  010346                    RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
 4798  021122  013703  002414                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4799  021126  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4800  021130  016663  000002  002250            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4801  021136  162763  000004  002250            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4802  021144  010337  002414                    MOV     R3,SSINDX       ;STORE IT BACK
 4803  021150  010046                            MOV     R0,-(SP)
 4804  021152  010146                            MOV     R1,-(SP)
 4805  021154  010446                            MOV     R4,-(SP)
 4806  021156  012737  000002  002430            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4807  021164  012701  000050                    MOV     #40.,R1         ;SET HEADER COUNT
 4808  021170  052737  100000  002416            BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
 4809  021176  012703  003256                    MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
 4810  021202  013704  002440                    MOV     RLBAS,R4        ;GET BASE ADDRESS
 4811  021206  062704  000006                    ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
 4812  021212  012737  000010  002446            MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
 4813  021220  053737  002444  002446            BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
 4814  021226  042737  002000  002446            BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4815  021234  005037  002450                    CLR     L.BA            ;CLEAR BA
 4816  021240  005037  002452                    CLR     L.DA            ;CLEAR DA
 4817  021244  005737  002524                    TST     DESHD           ;TEST IF HEAD 0
 4818  021250  001403                            BEQ     3$              ;YES - SKIP
 4819  021252  052737  000020  002452            BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
 4820  021260  013762  002452  000004  3$:       MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
 4821  021266  013762  002450  000002            MOV     L.BA,RLBA(R2)   ;LOAD RLBA
 4822  021274  032762  000200  000000            BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
 4823  021302  001003                            BNE     6$              ;YES - SKIP
 4824  021304  004737  016664                    JSR     PC,RDYCHK       ;ELSE CHECK READY
 4825  021310  021422                            #65$
 4826  021312  013762  002446  000000  6$:       MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
 4827  021320  012700  077777                    MOV     #77777,R0       ;SET COUNT FOR WAIT
 4828  021324  032762  000200  000000  7$:       BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
 4829  021332  001015                            BNE     8$              ;YES - SKIP
 4830  021334  005300                            DEC     R0              ;DEC COUNT
 4831  021336  001372                            BNE     7$              ;SKIP IF NOT YET 0
```

```
 4832  021340  004737  014530                 JSR     PC,READRL       ;ELSE GET ALL REGISTERS
 4833  021344  004737  014562                 JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
 4834  021350  012603                         MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 4835  021352                                 ERRHRD  10025.,,ERR1
  (3)  021352  104443                         TRAP    T$ERCODE
  (5)  021354  023451                         .WORD   10025
  (5)  021356  011344                         .WORD   ERR1
 4836  021360  005037  002430                 CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4837  021364  000416                         BR      65$
 4838  021366  005737  002456          8$:    TST     T.CS            ;TEST FOR ANY ERRORS
 4839  021372  100006                         BPL     12$             ;NO - SKIP
 4840  021374                                 ERRHRD  10026.,,ERR6
  (3)  021374  104443                         TRAP    T$ERCODE
  (5)  021376  023452                         .WORD   10026
  (5)  021400  011646                         .WORD   ERR6
 4841  021402  005037  002430                 CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4842  021406  000405                         BR      65$
 4843  021410  011423                 12$:    MOV     (R4),(R3)+      ;STORE HEADER WORDS
 4844  021412  011423                         MOV     (R4),(R3)+
 4845  021414  011423                         MOV     (R4),(R3)+
 4846  021416  005301                         DEC     R1              ;DEC HEADER COUNT
 4847  021420  001334                         BNE     6$
 4848  021422  162737  000002  002414  65$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4849  021430  012604                         MOV     (SP)+,R4        ;RESTORE REGISTERS
 4850  021432  012601                         MOV     (SP)+,R1
 4851  021434  012600                         MOV     (SP)+,R0
 4852  021436  012603                         MOV     (SP)+,R3
 4853  021440  005737  002430                 TST     ERRSWI          ;TEST IF ERROR RETURN
 4854  021444  001403                         BEQ     99$             ;YES - SKIP
 4855  021446  063716  002430                 ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4856  021452  000207                         RTS     PC
 4857  021454  017616  000000          99$:   MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4858  021460  000207                         RTS     PC
 4859
 4860
 4862                                  ;      GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
 4863                                  ;      IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
 4864                                  ;      IN OBUFF.
 4865  021462  010146                 DATGEN: MOV     R1,-(SP)        ;STORE REGISTERS
 4866  021464  010346                         MOV     R3,-(SP)
 4867  021466  010446                         MOV     R4,-(SP)
 4868  021470  012701  003656                 MOV     #OBUFF,R1       ;SET POINTER TO OBUFF
 4869  021474  012504                         MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
 4870  021476  006304                         ASL     R4              ;ADJUST IT FOR INDEXING
 4871  021500  016403  002224                 MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
 4872  021504  011321                         MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
 4873  021506  001421                         BEQ     5$              ;SKIP IF PATTERN IS 0
 4874  021510  021327  177777                 CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
 4875  021514  001416                         BEQ     5$              ;YES - SKIP
 4876  021516  020427  000010                 CMP     R4,#8.          ;TEST IF PATTERN 5
 4877  021522  001403                         BEQ     3$              ;YES - SKIP
 4878  021524  020427  000020                 CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
 4879  021530  002413                         BLT     6$              ;NO - SKIP
 4880  021532  005723                  3$:    TST     (R3)+           ;BUMP SOURCE POINTER
 4881  021534  012321                         MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
 4882  021536  012321                         MOV     (R3)+,(R1)+
```

```
 4883  021540  012704  000015                 MOV     #13.,R4          ;SET COUNT
 4884  021544  012703  003656                 MOV     #OBUFF,R3        ;RESET POINTER
 4885  021550  000406                         BR      8$
 4886  021552  012703  003656           5$:   MOV     #OBUFF,R3        ;ELSE SET OBUFF AS PATTERN SOURCE
 4887  021556  000401                         BR      7$               ;GO TO FILL
 4888  021560  005723                   6$:   TST     (R3)+            ;BUMP SOURCE POINTER
 4889  021562  012704  000017           7$:   MOV     #15.,R4          ;SET MOVE COUNT
 4890  021566  012321                   8$:   MOV     (R3)+,(R1)+      ;MOVE 15 WORDS INTO BUFFER
 4891  021570  005304                         DEC     R4
 4892  021572  001375                         BNE     8$
 4893  021574  012703  003656                 MOV     #OBUFF,R3        ;SET SOURCE TO TOP OF OBUFF
 4894  021600  012704  000160                 MOV     #112.,R4         ;SET COUNT FOR REST OF BUFFER
 4895  021604  012321                   10$:  MOV     (R3)+,(R1)+      ;REPEAT PATTERN IN BUFFER
 4896  021606  005304                         DEC     R4
 4897  021610  001375                         BNE     10$
 4898  021612  012604                         MOV     (SP)+,R4         ;RESTORE REGISTERS
 4899  021614  012603                         MOV     (SP)+,R3
 4900  021616  012601                         MOV     (SP)+,R1
 4901  021620  000205                         RTS     R5               ;RETURN
 4902
 4903                                   ;     DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
 4904                                   ;     ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
 4905  021622  010346                   DATCOM: MOV     R3,-(SP)         ;STORE R3
 4906  021624  013703  002414                 MOV     SSINDX,R3        ;GET SUBROUTINE STACK INDEX
 4907  021630  005723                         TST     (R3)+            ;BUMP INDEX TO NEXT ENTRY
 4908  021632  016663  000002  002250         MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4909  021640  162763  000004  002250         SUB     #4,SUBSTK(R3)    ;ADJUST IT TO CALLING LOCATION
 4910  021646  010337  002414                 MOV     R3,SSINDX        ;STORE IT BACK
 4911  021652  010146                         MOV     R1,-(SP)         ;STORE OTHER REGISTERS
 4912  021654  010446                         MOV     R4,-(SP)
 4913  021656  010546                         MOV     R5,-(SP)
 4914  021660  052737  000001  002416         BIS     #DATACMP,OPFLAG  ;SET DATA COMPARE FLAG
 4915  021666  005037  002426                 CLR     MORECE           ;CLEAR MORE ERROR FLAG
 4916  021672  012705  003656                 MOV     #OBUFF,R5        ;SET POINTERS TO DATA FOR COMPARE
 4917  021676  012704  003256                 MOV     #IBUFF,R4
 4918  021702  012703  000001                 MOV     #1,R3            ;SET WORD COUNTER
 4919  021706  012701  000200                 MOV     #128.,R1         ;SET COMPARE COUNT
 4920  021712  022425                   5$:   CMP     (R4)+,(R5)+      ;COMPARE DATA
 4921  021714  001052                         BNE     10$              ;ERROR - SKIP TO REPORT
 4922  021716  005203                   7$:   INC     R3               ;BUMP WORD COUNT
 4923  021720  005301                         DEC     R1               ;DEC COMPARE COUNT
 4924  021722  001373                         BNE     5$               ;LOOP IF NOT 0
 4925  021724  042737  000001  002416   9$:   BIC     #DATACMP,OPFLAG  ;CLEAR DATA COMPARE FLAG
 4926  021732  005737  002430                 TST     ERRSWI           ;TEST IF ANY COMPARE ERRORS
 4927  021736  001021                         BNE     15$              ;NO - SKIP
 4928  021740  012701  000200                 MOV     #128.,R1         ;SET REPORT VALUE
 4929  021744                                 PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
(11)   021744  010146                         MOV     R1,-(SP)
(10)   021746  012746  010211                 MOV     #RESE6,-(SP)
 (9)   021752  013746  002426                 MOV     MORECE,-(SP)
 (8)   021756  012746  007310                 MOV     #TCERR,-(SP)
 (7)   021762  012746  011314                 MOV     #FMT27,-(SP)
 (6)   021766  012746  000005                 MOV     #5,-(SP)
 (3)   021772  010600                         MOV     SP,R0
 (4)   021774  104014                         EMT     C$PNTB
 (4)   021776  062706  000014                 ADD     #14,SP
```

```
 4930  022002  162737  000002  002414  15$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4931  022010  012605                          MOV     (SP)+,R5        ;RESTORE REGS
 4932  022012  012604                          MOV     (SP)+,R4
 4933  022014  012601                          MOV     (SP)+,R1
 4934  022016  012603                          MOV     (SP)+,R3
 4935  022020  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4936  022024  001403                          BEQ     99$             ;YES - SKIP
 4937  022026  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4938  022032  000207                          RTS     PC
 4939  022034  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4940  022040  000207                          RTS     PC
 4941  022042  023737  002426  013174  10$:    CMP     MORECE,DCLIMW   ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
 4942  022050  002010                          BGE     13$             ;YES - SKIP
 4943  022052  024445                          CMP     -(R4),-(R5)     ;SET PTRS BACK TO ERROR WORDS
 4944  022054                                  ERRHRD  10035.,,ERR10   ;REPORT ERROR
  (3)  022054  104443                          TRAP    T$ERCODE
  (5)  022056  023463                          .WORD   10035
  (5)  022060  012734                          .WORD   ERR10
 4945  022062  005037  002430                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
 4946  022066  022425                          CMP     (R4)+,(R5)+     ;BUMP PTRS PAST ERROR WORDS
 4947  022070  000712                          BR      7$              ;DO NEXT COMPARE
 4948  022072  005237  002426          13$:    INC     MORECE          ;BUMP ERROR COUNTER
 4949  022076  000707                          BR      7$              ;DO NEXT COMPARE
 4950
 4951                                  ;       WRITE AND READ DATA ROUTINE.
 4952  022100  012737  177777  002532  XWRITT: MOV     #-1,TEMP1       ;SET SPECIAL WRITE FOR TIMING FLAG
 4953  022106  000402                          BR      XWRIT1
 4954  022110  005037  002532          XWRITE: CLR     TEMP1           ;CLEAR SPECIAL WRITE FLAG
 4955  022114  012737  000112  002546  XWRIT1: MOV     #WTDATA,TEMP7   ;SET FOR WRITE
 4956  022122  022737  000377  002516          CMP     #255 ,CURCYL    ;TEST IF CYLINDER 255 (BAD SEC)
 4957  022130  001006                          BNE     1$              ;NO - SKIP
 4958  022132  005737  002524                  TST     DESHD           ;TEST IF HEAD 1 (BAD SECTOR FILES)
 4959  022136  001403                          BEQ     1$              ;NO - SKIP
 4960  022140  052737  004000  002416          BIS     #BADADD,OPFLAG  ;SET BAD ADDRESS FLAG
 4961  022146  000403                  1$:     BR      XREADG          ;SKIP TO EXECUTE
 4962  022150  012737  000114  002546  XREAD:  MOV     #RDDATA,TEMP7   ;SET FOR READ
 4963  022156  010346                  XREADG: MOV     R3,-(SP)        ;STORE R3
 4964  022160  013703  002414                  MOV     SSINDX,R3       ;SET SUBROUTINE INDEX
 4965  022164  005723                          TST     (R3)+           ;BUMP TO NEXT STACK ENTRY
 4966  022166  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4967  022174  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST TO POINT TO CALL
 4968  022202  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4969  022206  010046                          MOV     R0,-(SP)
 4970  022210  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
 4971  022212  010446                          MOV     R4,-(SP)
 4972  022214  004737  016664                  JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
 4973  022220  022570                          #65$
 4974  022222  012703  002446                  MOV     #L.CS,R3        ;GET ADDRESS OF LOAD REGS
 4975  022226  013713  002546                  MOV     TEMP7,(R3)      ;SET COMMAND
 4976  022232  053713  002444                  BIS     RLDRV,(R3)      ;INSERT DRIVE NUMBER
 4977  022236  042713  002000                  BIC     #BIT10,(R3)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4978  022242  032723  000004                  BIT     #BIT2,(R3)+     ;TEST IF WRITE DATA
 4979  022246  001403                          BEQ     3$              ;YES - SKIP
 4980  022250  012723  003256                  MOV     #IBUFF,(R3)+    ;ELSE SET BA FOR READ
 4981  022254  000402                          BR      4$
 4982  022256  012723  003656          3$:     MOV     #OBUFF,(R3)+    ;SET BA FOR WRITE
```

```
 4983  022262  013713  002516          4$:     MOV     CURCYL,(R3)     ;GET CURRENT CYLINDER
 4984  022266  012704  000007                  MOV     #7,R4           ;ALIGN IT IN DA
 4985  022272  006313                  5$:     ASL     (R3)
 4986  022274  005304                          DEC     R4
 4987  022276  001375                          BNE     5$
 4988  022300  005737  002524                  TST     DESHD           ;TEST IF HEAD 0
 4989  022304  001402                          BEQ     7$              ;YES - SKIP
 4990  022306  052713  000100                  BIS     #HSMSK,(R3)     ;SET FOR HEAD 1
 4991  022312  053723  002526          7$:     BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
 4992  022316  012713  177600                  MOV     #177600,(R3)    ;INSERT WORD COUNT
 4993  022322  005737  002532                  TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
 4994  022326  001402                          BEQ     8$              ;NO - SKIP
 4995  022330  012713  177777                  MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
 4996  022334  032737  004000  002416  8$:     BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
 4997  022342  001413                          BEQ     2$              ;NO - SKIP
 4998  022344  042737  173777  002416          BIC     # CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
 4999  022352  012703  010077                  MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
 5000  022356                                  ERRHRD  10032.,,ERR1
   (3) 022356  104443                          TRAP    T$ERCODE
   (5) 022360  023460                          .WORD   10032
   (5) 022362  011344                          .WORD   ERR1
 5001  022364  005037  002416                  CLR     OPFLAG          ;CLEAR ALL FLAGS
 5002  022370  000475                          BR      64$
 5003  022372  005037  002420          2$:     CLR     DONE            ;CLEAR INTERRUPT FLAG
 5004  022376  005737  002532                  TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
 5005  022402  001072                          BNE     65$             ;YES - DO NOT START WRITE
 5006  022404  011362  000006                  MOV     (R3),RLMP(R2)   ;LOAD RL REGS
 5007  022410  014362  000004                  MOV     -(R3),RLDA(R2)
 5008  022414  014362  000002                  MOV     -(R3),RLBA(R2)
 5009  022420  014362  000000                  MOV     -(R3),RLCS(R2)
 5010  022424                          10$:    WAITUS  #3000.          ;WAIT 300MS FOR INTERRUPT
   (3) 022424  012700  005670                  MOV     #3000.,R0
   (3) 022430  104027                          EMT     C$WTU
 5011  022432  005737  002420                  TST     DONE            ;CHECK IF INTERRUPT
 5012  022436  001007                          BNE     14$             ;YES - SKIP
 5013  022440  004737  014562                  JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 5014  022444  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE
 5015  022446                                  ERRHRD  10030.,,ERR1
   (3) 022446  104443                          TRAP    T$ERCODE
   (5) 022450  023456                          .WORD   10030
   (5) 022452  011344                          .WORD   ERR1
 5016  022454  000443                          BR      64$
 5017  022456  032737  000001  002456  14$:    BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 5018  022464  001031                          BNE     20$             ;YES - SKIP
 5019  022466  012703  007333                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE
 5020  022472  012704  010300                  MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
 5021  022476                                  ERRHRD  10032.,,ERR5
   (3) 022476  104443                          TRAP    T$ERCODE
   (5) 022500  023460                          .WORD   10032
   (5) 022502  011576                          .WORD   ERR5
 5022  022504  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
 5023  022510  004737  014762          17$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
 5024  022514  022564                          64$
 5025  022516  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
 5026  022524  001011                          BNE     20$             ;YES - SKIP
 5027  022526  005301                          DEC     R1              ;DEC WAIT COUNT
```

```
 5028  022530  001367                        BNE     17$             ;LOOP IF NOT TIME DONE
 5029  022532  012704  010312                MOV     #C5SEC,R4       ;SET CONDITION 5 SECONDS
 5030  022536                                ERRHRD  10033.,,ERR5
  (3)  022536  104443                        TRAP    T$ERCODE
  (5)  022540  023461                        .WORD   10033
  (5)  022542  011576                        .WORD   ERR5
 5031  022544  005037  002430                CLR     ERRSWI          ;CLEAR ERROR SWITCH
 5032  022550  005737  002456          20$:  TST     T.CS            ;CHECK IF ANY ERROR
 5033  022554  100005                        BPL     65$             ;NO - SKIP
 5034  022556                                ERRHRD  10031.,,ERR6
  (3)  022556  104443                        TRAP    T$ERCODE
  (5)  022560  023457                        .WORD   10031
  (5)  022562  011646                        .WORD   ERR6
 5035  022564  005037  002430          64$:  CLR     ERRSWI          ;CLEAR ERROR SWITCH
 5036  022570  162737  000002  002414  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 5037  022576  012604                        MOV     (SP)+,R4        ;RESTORE REGISTERS
 5038  022600  012601                        MOV     (SP)+,R1
 5039  022602  012600                        MOV     (SP)+,R0
 5040  022604  012603                        MOV     (SP)+,R3
 5041  022606  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
 5042  022612  001403                        BEQ     99$             ;YES - SKIP
 5043  022614  063716  002430                ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
 5044  022620  000207                        RTS     PC
 5045  022622  017616  000000          99$:  MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
 5046  022626  000207                        RTS     PC
 5047
 5048                                  ;     BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
 5049                                  ;     DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
 5050  022630  010046                  BSCHK:MOV     R0,-(SP)        ;STORE REGISTERS
 5051  022632  010146                        MOV     R1,-(SP)
 5052  022634  010346                        MOV     R3,-(SP)
 5053  022636  005037  002432                CLR     BSFLAG          ;CLEAR FLAG
 5054  022642  012703  003062                MOV     #FBSFIL,R3      ;GET POINTER TO FACTORY FILE
 5055  022646  022713  177777                CMP     #-1,(R3)        ;CHECK IF ALL ONES
 5056  022652  001005                        BNE     4$              ;NO SKIP TO TEST
 5057  022654  012703  002666          2$:   MOV     #SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILE
 5058  022660  022713  177777                CMP     #-1,(R3)        ;CHECK IF ALL ONES
 5059  022664  001431                        BEQ     20$             ;YES - EXIT
 5060  022666  013700  002514          4$:   MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
 5061  022672  012701  000007                MOV     #7,R1           ;POSITION CYLINDER
 5062  022676  006300                  5$:   ASL     R0
 5063  022700  005301                        DEC     R1
 5064  022702  001375                        BNE     5$
 5065  022704  005737  002524                TST     DESHD           ;CHECK IF HEAD 0
 5066  022710  001402                        BEQ     7$              ;YES - SKIP
 5067  022712  052700  000100                BIS     #BIT6,R0        ;INSERT HEAD 1
 5068  022716  053700  002526          7$:   BIS     DESSEC,R0       ;INSERT SECTOR
 5069  022722  022300                  8$:   CMP     (R3)+,R0        ;CHECK THIS WORD IN FILE
 5070  022724  001402                        BEQ     12$             ;YES - EXIT,ERROR
 5071  022726  101005                        BHI     15$             ;EXIT- NO ERROR
 5072  022730  000774                        BR      8$
 5073  022732  012737  000001  002432  12$:  MOV     #1,BSFLAG       ;SET ERROR FLAG
 5074  022740  000403                        BR      20$             ;GO TO EXIT
 5075  022742  020327  003062          15$:  CMP     R3,#FBSFIL      ;DONE BOTH FILES?
 5076  022746  003342                        BGT     2$              ;NO GO DO SOFTWARE FILE
 5077  022750  012603                  20$:  MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
```

```
 5078  022752  012601                        MOV     (SP)+,R1
 5079  022754  012600                        MOV     (SP)+,R0
 5080  022756  005737  002432                TST     BSFLAG          ;CHECK IF ERROR
 5081  022762  001003                        BNE     99$             ;YES - SKIP
 5082  022764  062716  000002                ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
 5083  022770  000207                        RTS     PC
 5084  022772  017616  000000        99$:    MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
 5085  022776  000207                        RTS     PC
 5086
 5088                          ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
 5089                          ;       OPERATION BEING PERFORMED PORTION OF ALL
 5090                          ;       ERROR MESSAGES.
 5091  023000  010446          RPTOP:  MOV     R4,-(SP)
 5092  023002  005737  002414                TST     SSINDX          ;TEST SUBROUTINE INDEX 0
 5093  023006  001433                        BEQ     1$              ;SKIP IF 0
 5094  023010  012704  000002                MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
 5095  023014                                PRINTB  #FMT9,#SEQMES   ;PRINT "SUBROUTINE CALL SEQ"
  (8)  023014  012746  007156                MOV     #SEQMES,-(SP)
  (7)  023020  012746  010633                MOV     #FMT9,-(SP)
  (6)  023024  012746  000002                MOV     #2,-(SP)
  (3)  023030  010600                        MOV     SP,R0
  (4)  023032  104014                        EMT     C$PNTB
  (4)  023034  062706  000006                ADD     #6,SP
 5096  023040                        3$:     PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
  (8)  023040  016446  002250                MOV     SUBSTK(R4),-(SP)
  (7)  023044  012746  011006                MOV     #FMT16,-(SP)
  (6)  023050  012746  000002                MOV     #2,-(SP)
  (3)  023054  010600                        MOV     SP,R0
  (4)  023056  104014                        EMT     C$PNTB
  (4)  023060  062706  000006                ADD     #6,SP
 5097  023064  062704  000002                ADD     #2,R4           ;BUMP INDEX
 5098  023070  020437  002414                CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
 5099  023074  003761                        BLE     3$              ;LOOP IF NOT ALL PRINTED YET
 5100  023076                        1$:     PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
  (9)  023076  012746  006007                MOV     #TSTLAB,-(SP)
  (8)  023102  013746  002424                MOV     ERHEAD,-(SP)
  (7)  023106  012746  010436                MOV     #FMT4,-(SP)
  (6)  023112  012746  000003                MOV     #3,-(SP)
  (3)  023116  010600                        MOV     SP,R0
  (4)  023120  104014                        EMT     C$PNTB
  (4)  023122  062706  000010                ADD     #10,SP
 5101  023126  042737  030000  002416        BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
 5102  023134  013701  002446                MOV     L.CS,R1         ;GET COMMAND EXECUTED
 5103  023140  042701  177741                BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
 5104  023144  022701  000006                CMP     #6,R1   ;TEST IF SEEK OPERATION
 5105  023150  001003                        BNE     2$              ;NO - SKIP
 5106  023152  052737  010000  002416        BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
 5107  023160  022701  000012        2$:     CMP     #12,R1          ;TEST IF WRITE
 5108  023164  001003                        BNE     20$             ;NO - SKIP
 5109  023166  052737  020000  002416        BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 5110  023174  022701  000014        20$:    CMP     #14,R1          ;TEST IF READ
 5111  023200  001003                        BNE     22$             ;NO - SKIP
 5112  023202  052737  020000  002416        BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 5113  023210                        22$:    PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
  (9)  023210  016146  002112                MOV     OPMSGS(R1),-(SP)
  (8)  023214  012746  004733                MOV     #MOPER,-(SP)
```

```
  (7)  023220  012746  010414                MOV     #FMT1,-(SP)
  (6)  023224  012746  000003                MOV     #3,-(SP)
  (3)  023230  010600                        MOV     SP,R0
  (4)  023232  104014                        EMT     C$PNTB
  (4)  023234  062706  000010                ADD     #10,SP
5114   023240  020127  000004                CMP     R1,#4           ;CHECK IF GET STATUS
5115   023244  001007                        BNE     4$              ;NO - SKIP
5116   023246  032737  000010  002452        BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
5117   023254  001403                        BEQ     4$              ;NO - SKIP
5118   023256  012701  000016                MOV     #16,R1          ;SET TO PRINT WITH RESET
5119   023262  000436                        BR      9$
5120   023264  032737  007777  002416  4$:   BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
5121   023272  001424                        BEQ     8$              ;NO - SKIP
5122   023274  013704  002416                MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
5123   023300  012701  000020                MOV     #20,R1          ;PRESET THE POINTER
5124   023304  032704  000001          5$:   BIT     #BIT00,R4       ;CHECK THE BIT
5125   023310  001003                        BNE     6$              ;IF SET - SKIP
5126   023312  005721                        TST     (R1)+           ;BUMP POINTER
5127   023314  006204                        ASR     R4
5128   023316  000772                        BR      5$
5129   023320                         6$:    PRINTB  #FMT2,OPMSGS(R1)
  (8)  023320  016146  002112                MOV     OPMSGS(R1),-(SP)
  (7)  023324  012746  010430                MOV     #FMT2,-(SP)
  (6)  023330  012746  000002                MOV     #2,-(SP)
  (3)  023334  010600                        MOV     SP,R0
  (4)  023336  104014                        EMT     C$PNTB
  (4)  023340  062706  000006                ADD     #6,SP
5130   023344  032737  100000  002416  8$:   BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
5131   023352  001415                        BEQ     10$             ;NO - SKIP
5132   023354  012701  000050                MOV     #50,R1          ;ELSE PRINT IT
5133   023360                         9$:    PRINTB  #FMT2,OPMSGS(R1)
  (8)  023360  016146  002112                MOV     OPMSGS(R1),-(SP)
  (7)  023364  012746  010430                MOV     #FMT2,-(SP)
  (6)  023370  012746  000002                MOV     #2,-(SP)
  (3)  023374  010600                        MOV     SP,R0
  (4)  023376  104014                        EMT     C$PNTB
  (4)  023400  062706  000006                ADD     #6,SP
5134   023404  000434                        BR      15$             ;SKIP
5135   023406  032737  010000  002416  10$:   BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
5136   023414  001430                        BEQ     15$             ;NO - SKIP
5137   023416                                 PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
  (15) 023416  013746  002524                MOV     DESHD,-(SP)
  (14) 023422  012746  007117                MOV     #HDWD,-(SP)
  (13) 023426  013746  002522                MOV     DESSGN,-(SP)
  (12) 023432  012746  007112                MOV     #SGNWD,-(SP)
  (11) 023436  013746  002520                MOV     DESDIF,-(SP)
  (10) 023442  012746  007104                MOV     #DIFWD,-(SP)
  (9)  023446  013746  002512                MOV     OLDCYL,-(SP)
  (8)  023452  012746  007135                MOV     #FRMWD,-(SP)
  (7)  023456  012746  010654                MOV     #FMT13,-(SP)
  (6)  023462  012746  000011                MOV     #11,-(SP)
  (3)  023466  010600                        MOV     SP,R0
  (4)  023470  104014                        EMT     C$PNTB
  (4)  023472  062706  000024                ADD     #24,SP
5138   023476  032737  020000  002416  15$:   BIT     #RORWOP,OPFLAG  ;TEST IF READ OR WRITE SET
5139   023504  001424                        BEQ     17$             ;NO - SKIP
```

```
 5140  023506                                 PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
  (13) 023506  013746  002526                 MOV     DESSEC,-(SP)
  (12) 023512  012746  007123                 MOV     #SECWD,-(SP)
  (11) 023516  013746  002524                 MOV     DESHD,-(SP)
  (10) 023522  012746  007117                 MOV     #HDWD,-(SP)
   (9) 023526  013746  002516                 MOV     CURCYL,-(SP)
   (8) 023532  012746  007130                 MOV     #CYLWD,-(SP)
   (7) 023536  012746  011203                 MOV     #FMT22,-(SP)
   (6) 023542  012746  000007                 MOV     #7,-(SP)
   (3) 023546  010600                         MOV     SP,R0
   (4) 023550  104014                         EMT     C$PNTB
   (4) 023552  062706  000020                 ADD     #20,SP
 5141  023556  004737  024230          17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
 5142  023562  012604                         MOV     (SP)+,R4        ;RESTORE R4
 5143  023564  000207                         RTS     PC
 5144
 5145                                 ;       REPORT REASON ROUTINE
 5146                                 ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
 5147  023566  010146                 RPTRES: MOV     R1,-(SP)        ;STORE R1
 5148  023570  010346                         MOV     R3,-(SP)        ;STORE R3
 5149  023572  010446                         MOV     R4,-(SP)        ;STORE R4
 5150  023574  012701  002474                 MOV     #RESPARM,R1     ;GET START OF PARAM
 5151  023600  012103                         MOV     (R1)+,R3        ;GET NUMBER OF PARAM
 5152  023602                                 PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
   (9) 023602  011146                         MOV     (R1),-(SP)
   (8) 023604  012746  004747                 MOV     #MRSLT,-(SP)
   (7) 023610  012746  010421                 MOV     #FMT1.1,-(SP)
   (6) 023614  012746  000003                 MOV     #3,-(SP)
   (3) 023620  010600                         MOV     SP,R0
   (4) 023622  104014                         EMT     C$PNTB
   (4) 023624  062706  000010                 ADD     #10,SP
 5153  023630  021127  007731                 CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
 5154  023634  001453                         BEQ     6$              ;YES - SKIP REST OF REPORT
 5155  023636  012704  010640                 MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
 5156  023642  022127  007724                 CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
 5157  023646  001002                         BNE     3$              ;NO - SKIP
 5158  023650  012704  010646                 MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
 5159  023654  005303                  3$:    DEC     R3              ;DEC PARAM COUNT
 5160  023656  001442                         BEQ     6$              ;IF 0 - EXIT
 5161  023660                                 PRINTB  R4,#RESE3,(R1)+  ;REPORT IS VALUE
   (9) 023660  012146                         MOV     (R1)+,-(SP)
   (8) 023662  012746  010173                 MOV     #RESE3,-(SP)
   (7) 023666  010446                         MOV     R4,-(SP)
   (6) 023670  012746  000003                 MOV     #3,-(SP)
   (3) 023674  010600                         MOV     SP,R0
   (4) 023676  104014                         EMT     C$PNTB
   (4) 023700  062706  000010                 ADD     #10,SP
 5162  023704                                 PRINTB  R4,#RESE4,(R1)+  ;REPORT SB VALUE
   (9) 023704  012146                         MOV     (R1)+,-(SP)
   (8) 023706  012746  010177                 MOV     #RESE4,-(SP)
   (7) 023712  010446                         MOV     R4,-(SP)
   (6) 023714  012746  000003                 MOV     #3,-(SP)
   (3) 023720  010600                         MOV     SP,R0
   (4) 023722  104014                         EMT     C$PNTB
   (4) 023724  062706  000010                 ADD     #10,SP
 5163  023730  162703  000002                 SUB     #2,R3           ;DEC PARAM COUNT
```

```
 5164  023734  001413                        BEQ     6$                  ;IF 0 - EXIT
 5165  023736                                PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
  (9)  023736  012146                        MOV     (R1)+,-(SP)
  (8)  023740  012746  010204                MOV     #RESE5,-(SP)
  (7)  023744  012746  010414                MOV     #FMT1,-(SP)
  (6)  023750  012746  000003                MOV     #3,-(SP)
  (3)  023754  010600                        MOV     SP,R0
  (4)  023756  104014                        EMT     C$PNTB
  (4)  023760  062706  000010                ADD     #10,SP
 5166  023764  012604                6$:     MOV     (SP)+,R4            ;RESTORE REGS
 5167  023766  012603                        MOV     (SP)+,R3
 5168  023770  012601                        MOV     (SP)+,R1
 5169  023772  000207                        RTS     PC                  ;RETURN
 5170
 5171                                ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
 5172                                ;       AND ALL REGISTER CONTENTS.
 5173  023774                        RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11) 023774  005046                        CLR     -(SP)
  (11) 023776  153716  002445                BISB    RLDRV+1,(SP)
  (10) 024002  012746  005423                MOV     #DRVNAM,-(SP)
  (9)  024006  013746  002440                MOV     RLBAS,-(SP)
  (8)  024012  012746  005412                MOV     #BASADD,-(SP)
  (7)  024016  012746  010447                MOV     #FMT5,-(SP)
  (6)  024022  012746  000005                MOV     #5,-(SP)
  (3)  024026  010600                        MOV     SP,R0
  (4)  024030  104014                        EMT     C$PNTB
  (4)  024032  062706  000014                ADD     #14,SP
 5174                                ;       REPORT RL11 REGISTERS
 5175  024036                                PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
  (13) 024036  012746  007117                MOV     #HDWD,-(SP)
  (12) 024042  012746  007130                MOV     #CYLWD,-(SP)
  (11) 024046  012746  005542                MOV     #MPNAM,-(SP)
  (10) 024052  012746  005530                MOV     #BANAM,-(SP)
  (9)  024056  012746  005535                MOV     #DANAM,-(SP)
  (8)  024062  012746  005523                MOV     #CSNAM,-(SP)
  (7)  024066  012746  010467                MOV     #FMT6,-(SP)
  (6)  024072  012746  000007                MOV     #7,-(SP)
  (3)  024076  010600                        MOV     SP,R0
  (4)  024100  104014                        EMT     C$PNTB
  (4)  024102  062706  000020                ADD     #20,SP
 5176  024106                                PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
  (12) 024106  013746  002454                MOV     L.MP,-(SP)
  (11) 024112  013746  002450                MOV     L.BA,-(SP)
  (10) 024116  013746  002452                MOV     L.DA,-(SP)
  (9)  024122  013746  002446                MOV     L.CS,-(SP)
  (8)  024126  012746  005547                MOV     #LAB1,-(SP)
  (7)  024132  012746  010601                MOV     #FMT8,-(SP)
  (6)  024136  012746  000006                MOV     #6,-(SP)
  (3)  024142  010600                        MOV     SP,R0
  (4)  024144  104014                        EMT     C$PNTB
  (4)  024146  062706  000016                ADD     #16,SP
 5177  024152                                PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
  (14) 024152  013746  002524                MOV     DESHD,-(SP)
  (13) 024156  013746  002516                MOV     CURCYL,-(SP)
  (12) 024162  013746  002464                MOV     T.MP,-(SP)
  (11) 024166  013746  002460                MOV     T.BA,-(SP)
```

```
   (10) 024172  013746  002462                    MOV     T.DA,-(SP)
   (9)  024176  013746  002456                    MOV     T.CS,-(SP)
   (8)  024202  012746  005562                    MOV     #LAB2,-(SP)
   (7)  024206  012746  010531                    MOV     #FMT7,-(SP)
   (6)  024212  012746  000010                    MOV     #10,-(SP)
   (3)  024216  010600                            MOV     SP,R0
   (4)  024220  104014                            EMT     C$PNTB
   (4)  024222  062706  000022                    ADD     #22,SP
  5178  024226  000207                            RTS     PC
  5179
  5180                                    ;       CLEAR PARAMETER BLOCK FOR REPORTING
  5181  024230  010546                    CLRPARM:        MOV     R5,-(SP)        ;STORE R5
  5182  024232  012701  002474                    MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
  5183  024236  012705  000005                    MOV     #5,R5           ;SET COUNT
  5184  024242  005021                    2$:     CLR     (R1)+           ;CLEAR WORD
  5185  024244  005305                            DEC     R5              ;DEC COUNT
  5186  024246  001375                            BNE     2$              ;LOOP UNTIL 0
  5187  024250  012701  002474                    MOV     #RESPARM,R1     ;RESET POINTER
  5188  024254  012605                            MOV     (SP)+,R5        ;RESTORE R5
  5189  024256  000207                            RTS     PC
  5190
  5191  024260                            ENDMOD
  5192
```

```
 5194  024260                          BGNMOD  HRDWTST
 5195                                  .SBTTL  *TEST 1          **DIFFERENCE OF 1 SEEK (PART 1)
 5196  024260                          BGNTST                   ;TEST 1
  (3)  024260                                                                          T1::
 5197  024260  012737  006015  002424          MOV     #P2T01E,ERHEAD  ;SET ERROR HEADER
 5198  024266  012737  000004  002530          MOV     #4,TEMP0        ;SET PASS COUNT
 5199  024274  004737  014714                  JSR     PC,TSTINT       ;INITIALIZE TEST
 5200  024300  004737  014732                  JSR     PC,GSTATR       ;GET STATUS
 5201  024304  024600                          #T1765$
 5202  024306  012737  177777  002534          MOV     #-1,TEMP2       ;SET -1 INTO DIFF AUGMENT FOR -1 SEEK
 5203  024314  012704  002516                  MOV     #CURCYL,R4      ;SET POINTERS
 5204  024320  012705  002514                  MOV     #NEWCYL,R5
 5205  024324  004737  017124                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
 5206  024330                          T172$:
 5207  024330                          BGNSUB
  (3)  024330                                                                          T1.1:
  (3)  024330  104002                          EMT     C$BSUB
 5208  024332  004737  020652                  JSR     PC,GETPOS       ;GET POSITION
 5209  024336  024536                          #60$
 5210  024340                                  INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  024340  104020                          EMT     C$INLP
 5211  024342                                  BNCOMPLETE      3$      ;NO - SKIP
  (2)  024342  103005                          BCC     3$
 5212  024344  021415                          CMP     (R4),(R5)       ;CHECK IF CURRENT = NEW
 5213  024346  001005                          BNE     4$              ;NO - SKIP
 5214  024350  004737  017210                  JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
 5215  024354  000421                          BR      9$              ;SKIP TO SEEK
 5216  024356  005437  002534          3$:     NEG     TEMP2           ;CHANGE DIFF AUGMENT FOR OPPOSITE DIR
 5217  024362  011415                  4$:     MOV     (R4),(R5)       ;MOV     CURRENT INTO OLD
 5218  024364  022714  000377                  CMP     #255.,(R4)      ;CHECK IF CURRENT AT 255
 5219  024370  001004                          BNE     7$              ;NO - SKIP
 5220  024372  012737  177777  002534          MOV     #-1,TEMP2       ;AT MAX CYL, MAKE NEXT SEEK REV
 5221  024400  000405                          BR      8$              ;SKIP
 5222  024402  005714                  7$:     TST     (R4)            ;TEST IF CURRENT AT 0
 5223  024404  001003                          BNE     8$              ;NO - SKIP
 5224  024406  012737  000001  002534          MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
 5225  024414  063715  002534          8$:     ADD     TEMP2,(R5)      ;ADD DIFF TO NEW CYL (+1 OR -1)
 5226  024420  004737  015624          9$:     JSR     PC,XSEEK        ;DO SEEK
 5227  024424  024536                          #60$
 5228  024426  004737  014762                  JSR     PC,GSTAT        ;GET STATUS
 5229  024432  024536                          #60$
 5230
 5231  024434  012703  000004                  MOV     #4,R3           ;SET EXPECTED STATE
 5232  024440  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE COUNT
 5233  024444  001404                          BEQ     10$             ;YES-SKIP
 5234  024446                                  ERRHRD  101.,,ERR7      ;REPORT STATE ERROR
  (3)  024446  104443                          TRAP    T$ERCODE
  (5)  024450  000145                          .WORD   101
  (5)  024452  012524                          .WORD   ERR7
 5235  024454  000423                          BR      16$             ;EXIT TEST
 5236  024456  012703  000005          10$:    MOV     #5,R3           ;SET EXPECTED STATE
 5237  024462  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 MS
 5238  024466  004737  014762          12$:    JSR     PC,GSTAT        ;GET STATUS
 5239  024472  024536                          #60$
 5240  024474  020337  002472                  CMP     R3,T.STAT       ;IS STATE 5?
 5241  024500  001411                          BEQ     16$             ;YES-SKIP
```

```
 5242  024502  005301                          DEC     R1              ;DEC WAIT COUNT
 5243  024504  001404                          BEQ     14$             ;SKIP IF 0
 5244  024506                                  WAITUS  #1
  (3)  024506  012700  000001                  MOV     #1,R0
  (3)  024512  104027                          EMT     C$WTU
 5245  024514  000764                          BR      12$
 5246  024516                          14$:    ERRHRD  102.,,ERR7      ;REPORT STATE ERROR
  (3)  024516  104443                          TRAP    T$ERCODE
  (5)  024520  000146                          .WORD   102
  (5)  024522  012524                          .WORD   ERR7
 5247  024524  012701  000062          16$:    MOV     #50.,R1         ;SET WAIT COUNT FOR 5 MS
 5248  024530  004737  020404                  JSR     PC,RDYWAIT      ;GO WAIT FOR DRIVE READY
 5249  024534  024536                          #60$
 5250  024536  012737  000002  002430  60$:    MOV     #2,ERRSW1       ;INIT ERROR SWITCH
 5251  024544                          ENDSUB
  (3)  024544                          L10021:
  (3)  024544  104003                          EMT     C$ESUB
 5252  024546                                  ESCAPE  TST             ;EXIT TEST IF ERROR
  (3)  024546  104010                          EMT     C$ESCAPE
  (3)  024550  000030                          .WORD   L10020-.
 5253  024552  005337  002530                  DEC     TEMP0           ;DEC PASS COUNT
 5254  024556  001410                          BEQ     24$             ;SKIP IF 0-DONE
 5255
 5256  024560  032737  000001  002530          BIT     #BIT0,TEMP0     ;TEST IF PASS=2
 5257  024566  001003                          BNE     23$             ;NO-SKIP
 5258  024570  004737  017150                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 5259  024574  024600                          24$                     ;ABORT RETURN
 5260  024576  000654                  23$:    BR      T172$
 5261  024600                          24$:
 5262  024600                          T1765$:
 5263  024600                          ENDTST
  (3)  024600                          L10020:
  (3)  024600  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 2          **DIFFERENCE OF 1 SEEK (PART 2)
024602                                  BGNTST                   ;TEST 2
(3)  024602                                                                                  T2::
5269 024602  012737  006015  002424             MOV     #P2TO2E,ERHEAD  ;SET ERROR HEADER
5270 024610  012737  000004  002530             MOV     #4,TEMP0        ;SET PASS COUNT
5271 024616  004737  014714                     JSR     PC,TSTINT       ;INITIALIZE TEST
5272 024622  004737  014732                     JSR     PC,GSTATR       ;GET STATUS, CLEAR DRIVE
5273 024626  025070                             #T1865$
5274 024630  004737  017124                     JSR     PC,CHOSHD       ;GO CHOSE HEAD
5275 024634  012737  177777  002534             MOV     #-1,TEMP2       ;SET DIFF AUGMENT TO -1 (REVERSE)
5276 024642  012703  002514                     MOV     #NEWCYL,R3      ;GET ADDRESSES
5277 024646  012704  002516                     MOV     #CURCYL,R4
5278 024652  012705  002512                     MOV     #OLDCYL,R5
5279 024656                          T187$:
5280 024656                          BGNSUB
(3)  024656                                                                                  T2.1:
(3)  024656  104002                             EMT     C$BSUB
5281 024660  004737  020652                     JSR     PC,GETPOS       ;GET CURRENT POSITION
5282 024664  025026                             #60$
5283 024666                                     INLOOP                  ;CHECK IF IN ERROR LOOP
(3)  024666  104020                             EMT     C$INLP
5284 024670                                     BNCOMPLETE      3$      ;NO - SKIP
(2)  024670  103005                             BCC     3$
5285 024672  021413                             CMP     (R4),(R3)       ;CHECK IF CURRENT = NEW
5286 024674  001005                             BNE     4$              ;NO - SKIP
5287 024676  004737  017210                     JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
5288 024702  000421                             BR      9$              ;SKIP TO SEEK
5289 024704  005437  002534          3$:        NEG     TEMP2           ;CHANGE DIFF AUGMENT FOR OPPOSITE DIR
5290 024710  011413                  4$:        MOV     (R4),(R3)       ;MOV CURRENT INTO NEW
5291 024712  022714  000377                     CMP     #255.,(R4)      ;CHECK IF CURRENT AT 255
5292 024716  001004                             BNE     7$              ;NO - SKIP
5293 024720  012737  177777  002534             MOV     #-1,TEMP2       ;AT MAX CYL, ,AKE NEXT SEEK REV
5294 024726  000405                             BR      8$              ;SKIP
5295 024730  005714                  7$:        TST     (R4)            ;TEST IF CURRENT AT 0
5296 024732  001003                             BNE     8$              ;NO - SKIP
5297 024734  012737  000001  002534             MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
5298 024742  063713  002534          8$:        ADD     TEMP2,(R3)      ;ADD DIFF TO NEW CYL (+1 OR -1)
5299 024746  004737  015624          9$:        JSR     PC,XSEEK        ;DO SEEK
5300 024752  025026                             #60$
5301 024754  012701  000226                     MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
5302 024760  004737  020404                     JSR     PC,RDYWAIT      ;WAIT FOR READY
5303 024764  025026                             #60$
5304 024766  004737  020652                     JSR     PC,GETPOS       ;STORE POSITION
5305 024772  025026                             #60$
5306 024774  011501                             MOV     (R5),R1         ;GET OLD POSITION
5307 024776  161401                             SUB     (R4),R1         ;SUBTRACT FROM NEW POINTER (FORWARD)
5308 025000  005737  002522                     TST     DESSGN          ;CHECK IF SIGN FORWARD
5309 025004  001402                             BEQ     10$             ;YES-SKIP, ELSE SUB FOR SEEK REVERSE
5310 025006  011401                             MOV     (R4),R1         ;GET NEW CYLINDER
5311 025010  161501                             SUB     (R5),R1         ;SUBTRACT FROM OLD CYL
5312 025012  022701  000001          10$:       CMP     #1,R1           ;CHECK IF RESULT IS DIFFERENCE OF 1
5313 025016  001403                             BEQ     12$             ;YES-SKIP
5314 025020                                     ERRHRD  201.,,ERR8      ;ELSE REPORT ERROR
(3)  025020  104443                             TRAP    T$ERCODE
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:30  PAGE 84-3
DZRLDA.P11    04-NOV-77 11:51          *TEST 2          **DIFFERENCE OF 1 SEEK (PART 2)                    SEQ 0088

   (5)  025022  000311                              .WORD   201
   (5)  025024  012574                              .WORD   ERR8
 5315   025026                          12$:
 5316   025026  012737  000002  002430  60$:      MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 5317   025034                          ENDSUB
   (3)  025034                          L10023:
   (3)  025034  104003                              EMT     C$ESUB
 5318   025036                                      ESCAPE  TST             ;EXIT TEST IF ERROR
   (3)  025036  104010                              EMT     C$ESCAPE
   (3)  025040  000030                              .WORD   L10022-.
 5319   025042  005337  002530                      DEC     TEMPO           ;DEC PASS COUNT
 5320   025046  001410                              BEQ     30$             ;EXIT IF DONE
 5321
 5322   025050  032737  000001  002530              BIT     #BIT0,TEMPO     ;TEST IF PASS 1 OR 3
 5323   025056  001003                              BNE     20$             ;YES-SKIP
 5324   025060  004737  017150                      JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 5325   025064  025070                              30$                     ;ABORT RETURN
 5326   025066  000673                  20$:        BR      T187$           ;LOOP
 5327   025070                          30$:
 5328   025070                          T1865$:
 5329   025070                          ENDTST
   (3)  025070                          L10022:
   (3)  025070  104001                              EMT     C$ETST
```

```
                                        .SBTTL  *TEST 3         **OUTER GUARD BAND DETECTION
5334   025072                           BGNTST                  ;TEST 3
 (3)   025072                                                                                  T3::
5335   025072  012737  006034  002424           MOV     #P2TO3E,ERHEAD  ;SET ERROR HEADER
5336   025100  004737  014714                   JSR     PC,TSTINT       ;INITIALIZE TEST
5337   025104  004737  014732                   JSR     PC,GSTATR       ;CLEAR DRIVE
5338   025110  025302                           #T1965$
5339   025112  004737  017124                   JSR     PC,CHOSHD       ;GO CHOSE HEAD
5340   025116  005005                   T197$:  CLR     R5              ;CLEAR FOR POSITION TO 0
5341   025120  004737  016324                   JSR     PC,POSHDS       ;POSITION HEADS
5342   025124  025302                           #T1965$
5343   025126                           BGNSUB
 (3)   025126                                                                                  T3.1:
 (3)   025126  104002                           EMT     C$BSUB
5344   025130  012737  177777  002514           MOV     #-1,NEWCYL      ;SET FOR GUARD BAND SEEK
5345   025136  004737  015624                   JSR     PC,XSEEK        ;DO SEEK
5346   025142  025256                           #60$
5347   025144  012701  000003                   MOV     #3.,R1          ;SET WAIT COUNT FOR 3MS
5348   025150  032762  000001  000000  8$:      BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE READY
5349   025156  001413                           BEQ     9$              ;NO-SKIP
5350   025160  004737  014762                   JSR     PC,GSTAT        ;GET DRIVE STATUS
5351   025164  025256                           60$
5352   025166  012703  007333                   MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
5353   025172  012704  010246                   MOV     #C1OMS,R4       ;SET CONDITION MESSAGE PTR
5354   025176                                   ERRHRD  301.,,ERR4      ;REPORT READY ERROR
 (3)   025176  104443                           TRAP    T$ERCODE
 (5)   025200  000455                           .WORD   301
 (5)   025202  011526                           .WORD   ERR4
5355   025204  000424                           BR      60$             ;EXIT TEST
5356   025206  005301                   9$:     DEC     R1              ;DEC WAIT COUNT
5357   025210  001404                           BEQ     12$             ;SKIP IF 0
5358   025212                                   WAITUS  #10.
 (3)   025212  012700  000012                   MOV     #10.,R0
 (3)   025216  104027                           EMT     C$WTU
5359   025220  000753                           BR      8$              ;LOOP
5360   025222  012701  000226           12$:    MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
5361   025226  004737  020404                   JSR     PC,RDYWAIT      ;WAIT FOR READY & REPORT IF NOT READY
5362   025232  025256                           #60$
5363
5364   025234  004737  020652                   JSR     PC,GETPOS       ;GET POSITION
5365   025240  025256                           #60$
5366   025242  005737  002516                   TST     CURCYL          ;CHECK IF HEADS STILL AT 0
5367   025246  001403                           BEQ     15$             ;YES-SKIP
5368   025250                                   ERRHRD  302.,,ERR8      ;ELSE REPORT CYLINDER ERROR
 (3)   025250  104443                           TRAP    T$ERCODE
 (5)   025252  000456                           .WORD   302
 (5)   025254  012574                           .WORD   ERR8
5369   025256                           15$:
5370   025256  012737  000002  002430   60$:    MOV     #2,ERRSW1       ;INIT ERROR SWITCH
5371   025264                           ENDSUB
 (3)   025264                           L10025:
 (3)   025264  104003                           EMT     C$ESUB
5372   025266                                   ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)   025266  104010                           EMT     C$ESCAPE
```

```
  (3)  025270  000012                          .WORD   L10024-.
5373   025272  004737  017150                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
5374   025276  025302                          17$                     ;ABORT RETURN
5375   025300  000706                          BR      T197$           ;REDO TEST
5376   025302                          17$:
5377   025302                          T1965$:
5378   025302                          ENDTST
  (3)  025302                          L10024:
  (3)  025302  104001                          EMT     C$ETST
```

```
 5380
 5381
 5382                                   .SBTTL  *TEST 4          **INCREMENTAL FORWARD SEEK HEAD 0
 5383  025304                           BGNTST                   ;TEST 4
  (3)  025304                                                                              T4::
 5384  025304  012737  006060  002424           MOV     #P2T04E,ERHEAD  ;SET ERROR HEADER
 5385  025312  004737  014714                   JSR     PC,TSTINT       ;INITIALIZE TEST
 5386  025316  004737  014732                   JSR     PC,GSTATR       ;CLEAR DRIVE
 5387  025322  025512                           #T2065$
 5388  025324  004737  017124                   JSR     PC,CHOSHD       ;GO CHOSE HEAD
 5389  025330  005737  002524                   TST     DESHD           ;TEST IF THIS IS HEAD 0
 5390  025334  001402                           BEQ     2$              ;YES - SKIP
 5391  025336                           EXIT    TST                     ;ELSE EXIT TEST
  (3)  025336  104032                           EMT     C$EXIT
  (3)  025340  000152                           .WORD   L10026-.
 5392  025342  013705  013164          2$:     MOV     LOLIMW,R5       ;CLEAR TO POSITION HEADS TO LOLIMIT
 5393  025346  004737  016324                   JSR     PC,POSHDS       ;POSITION HEADS
 5394  025352  025512                           #T2065$
 5395  025354                           BGNSUB
  (3)  025354                                                                              T4.1:
  (3)  025354  104002                           EMT     C$BSUB
 5396  025356  004737  020652          T206$:  JSR     PC,GETPOS       ;GET POSITION
 5397  025362  025502                           #60$
 5398  025364                                   INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  025364  104020                           EMT     C$INLP
 5399  025366                                   BNCOMPLETE      5$      ;NO - SKIP
  (2)  025366  103007                           BCC     5$
 5400  025370  023737  002516  002514           CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
 5401  025376  001003                           BNE     5$              ;NO - SKIP
 5402  025400  004737  017210                   JSR     PC,ONSWAP       ;ELSE SWAP NEW AND OLD CYLINDERS
 5403  025404  000405                           BR      7$              ;SKIP
 5404  025406  013737  002516  002514  5$:     MOV     CURCYL,NEWCYL   ;PLACE CURRENT INTO NEW
 5405  025414  005237  002514                   INC     NEWCYL          ;BUMP FOR ONE CYLINDER SEEK
 5406  025420                           7$:
 5407  025420  004737  015624                   JSR     PC,XSEEK        ;DO SEEK
 5408  025424  025502                           #60$
 5409  025426  012701  000226                   MOV     #150.,R1        ;SET WAIT TIME 15 MS
 5410  025432  004737  020404                   JSR     PC,RDYWAIT      ;WAIT FOR READY
 5411  025436  025502                           #60$
 5412
 5413  025440  004737  021000                   JSR     PC,VERPOS       ;GO VERIFY POSITON
 5414  025444  025502                           #60$
 5415
 5416  025446  032737  000002  013162           BIT     #ALLSEC,MISWIW  ;TEST IF CHECK ALL SECTORS
 5417  025454  001406                           BEQ     11$             ;NO-SKIP
 5418  025456  004737  021120                   JSR     PC,RDALHD       ;GO READ ALL HEADERS
 5419  025462  025502                           #60$
 5420  025464  004737  020016                   JSR     PC,VERHDR       ;GO VERIFY HEADER
 5421  025470  025502                           #60$
 5422  025472                           11$:
 5423  025472  023737  013166  002514           CMP     HILIMW,NEWCYL   ;CHECK IF HILIMIT REACHED
 5424  025500  103726                           BLO     T206$           ;NO-LOOP
 5425  025502  012737  000002  002430  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 5426  025510                           ENDSUB
  (3)  025510                           L10027:
  (3)  025510  104003                           EMT     C$ESUB
```

```
 5427  025512                          T2065$:
 5428  025512                          ENDTST
  (3)  025512                          L10026:
  (3)  025512  104001                          EMT     C$ETST
```

```
5430
5431
5432                                    .SBTTL  *TEST 5          **INCREMENTAL REVERSE SEEK HEAD 0
5433  025514                            BGNTST                   ;TEST 5
 (3)  025514                                                                              T5::
5434  025514  012737  006102  002424            MOV     #P2TOSE,ERHEAD  ;SET ERROR HEADER
5435  025522  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
5436  025526  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
5437  025532  025722                            #T2165$
5438  025534  004737  017124                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
5439  025540  005737  002524                    TST     DESHD           ;TEST IF HEAD 0 SELECTED
5440  025544  001402                            BEQ     2$              ;YES - SKIP
5441  025546                            EXIT    TST                     ;ELSE EXIT TEST
 (3)  025546  104032                            EMT     C$EXIT
 (3)  025550  000152                            .WORD   L10030-.
5442  025552  013705  013166            2$:     MOV     HILIMW,R5       ;SET TO POSITION HDS TO HILIMIT
5443  025556  004737  016324                    JSR     PC,POSHDS       ;POSITION HEADS
5444  025562  025722                            #T2165$
5445  025564                            BGNSUB
 (3)  025564                                                                              T5.1:
 (3)  025564  104002                            EMT     C$BSUB
5446  025566  004737  020652            T216$:  JSR     PC,GETPOS       ;GET POSITION
5447  025572  025712                            #60$
5448  025574                                    INLOOP                  ;CHECK IF IN ERROR LOOP
 (3)  025574  104020                            EMT     C$INLP
5449  025576                                    BNCOMPLETE      5$      ;NO - SKIP
 (2)  025576  103007                            BCC     5$
5450  025600  023737  002516  002514            CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DES LOC
5451  025606  001003                            BNE     5$              ;NO - SKIP
5452  025610  004737  017210                    JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDERS
5453  025614  000405                            BR      7$              ;SKIP
5454  025616  013737  002516  002514    5$:     MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
5455  025624  005337  002514                    DEC     NEWCYL          ;DEC FOR ONE CYLINDER REVERSE SEEK
5456  025630  004737  015624            7$:     JSR     PC,XSEEK        ;SEEK TO IT
5457  025634  025712                            #60$
5458  025636  012701  000226                    MOV     #150.,R1        ;SET WAIT FOR 15 MS
5459  025642  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
5460  025646  025712                            #60$
5461
5462  025650  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
5463  025654  025712                            #60$
5464
5465  025656  032737  000002  002416            BIT     #ALLSEC,OPFLAG  ;TEST IF USE ALL SECTORS
5466  025664  001406                            BEQ     11$             ;NO-SKIP
5467  025666  004737  021120                    JSR     PC,RDALHD       ;ELSE READ ALL THE HDRS
5468  025672  025712                            #60$
5469  025674  004737  020016                    JSR     PC,VERHDR       ;VERIFY THE HEADERS
5470  025700  025712                            #60$
5471  025702                            11$:
5472  025702  023737  013164  002514            CMP     LOLIMW,NEWCYL   ;CHECK IF REACHED LOLIMIT
5473  025710  103726                            BLO     T216$           ;NO - LOOP
5474  025712  012737  000002  002430    60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5475  025720                            ENDSUB
 (3)  025720                            L10031:
 (3)  025720  104003                            EMT     C$ESUB
5476  025722                            T2165$:
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:30  PAGE 84-9
DZRLDA.P11     04-NOV-77 11:51          *TEST 5          **INCREMENTAL REVERSE SEEK HEAD 0                    SEQ 0094

  5477  025722                         ENDTST
   (3)  025722                         L10030:
   (3)  025722  104001                          EMT     CSETST
```

```
                                        .SBTTL  *TEST 6          **INCREMENTAL FORWARD SEEK HEAD 1
025724                                  BGNTST                   ;TEST 6
(3)  025724                                                                                  T6::
5483 025724 012737 006124 002424                MOV     #P2T06E,ERHEAD  ;SET ERROR HEADER
5484 025732 004737 014714                       JSR     PC,TSTINT       ;INITIALIZE TEST
5485 025736 004737 014732                       JSR     PC,GSTATR       ;CLEAR DRIVE
5486 025742 026146                              #T2265$
5487 025744 005037 002524                       CLR     DESHD           ;SET HEAD TO 0
5488 025750 013705 013164                       MOV     LOLIMW,R5       ;CLEAR FOR POSITION HDS TO LOLIMIT
5489 025754 004737 016324                       JSR     PC,POSHDS       ;POSITION HDS
5490 025760 026146                              #T2265$
5491 025762 012737 000001 002524                MOV     #1,DESHD        ;SET TO HEAD 1
5492 025770 032737 010000 013162                BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
5493 025776 001405                              BEQ     2$              ;NO - SKIP
5494 026000 005737 013170                       TST     HEADW           ;TEST IF IT IS HEAD 0
5495 026004 001002                              BNE     2$              ;NO - SKIP
5496 026006                                     EXIT    TST             ;ELSE EXIT TEST
(3)  026006 104032                              EMT     C$EXIT
(3)  026010 000136                              .WORD   L10032-.
5497 026012                             2$:
5498 026012                             BGNSUB
(3)  026012                                                                                  T6.1:
(3)  026012 104002                              EMT     C$BSUB
5499 026014 004737 020652               T227$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
5500 026020                                     INLOOP                  ;CHECK IF IN ERROR LOOP
(3)  026020 104020                              EMT     C$INLP
5501 026022                                     BNCOMPLETE 5$           ;NO - SKIP
(2)  026022 103007                              BCC     5$
5502 026024 023737 002516 002514                CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED LOCATION
5503 026032 001003                              BNE     5$              ;NO - SKIP
5504 026034 004737 017210                       JSR     PC,ONSWAP       ;SWAP OLD AND NEW CYLINDER
5505 026040 000405                              BR      7$              ;SKIP
5506 026042 013737 002516 002514        5$:     MOV     CURCYL,NEWCYL   ;MOVE CURRENT INTO NEW
5507 026050 005237 002514                       INC     NEWCYL          ;BUMP NEWCYL FOR ONE CYL FWRD SEEK
5508 026054                             7$:
5509 026054 004737 015624                       JSR     PC,XSEEK        ;DO SEEK
5510 026060 026136                              #60$
5511 026062 012701 000226                       MOV     #150.,R1        ;SET WAIT COUNT 15 MS
5512 026066 004737 020404                       JSR     PC,RDYWAIT      ;WAIT FOR READY
5513 026072 026136                              #60$
5514 026074 004737 021000                       JSR     PC,VERPOS       ;VERIFY POSITION IS CORRECT
5515 026100 026136                              #60$
5516
5517 026102 032737 000002 013162                BIT     #ALLSEC,MISWIW  ;CHECK IF USE ALL SECTORS
5518 026110 001406                              BEQ     9$              ;NO-SKIP
5519 026112 004737 021120                       JSR     PC,RDALHD       ;ELSE READ ALL HEADERS
5520 026116 026136                              #60$
5521 026120 004737 020016                       JSR     PC,VERHDR       ;VERIFY HEADERS
5522 026124 026136                              #60$
5523 026126                             9$:
5524 026126 023737 013166 002514                CMP     HILIMW,NEWCYL   ;CHECK IF DONE
5525 026134 101327                              BHI     T227$           ;NO - LOOP
5526 026136 012737 000002 002430        60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5527 026144                             ENDSUB
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:30  PAGE 84-11
DZRLDA.P11    04-NOV-77 11:51          *TEST 6          **INCREMENTAL FORWARD SEEK HEAD 1                    SEQ 0096

   (3)  026144                        L10033:
   (3)  026144  104003                        EMT     C$ESUB
 5528   026146                        T2265$:
 5529   026146                        ENDTST
   (3)  026146                        L10032:
   (3)  026146  104001                        EMT     C$ETST
```

```
5531
5532
5533                                    .SBTTL  *TEST 7          **INNER GUARD BAND DETECTION
5534  026150                            BGNTST                   ;TEST 7
 (3)  026150                                                                              T7::
5535  026150  012737  006146  002424            MOV     #P2TO7E,ERHEAD  ;SET ERROR HEADER
5536  026156  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
5537  026162  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
5538  026166  026346                            #T2365$
5539  026170  004737  017124                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
5540  026174  012705  000377            T233$:  MOV     #255.,R5        ;SET FOR POSITION TO 255.
5541  026200  004737  016324                    JSR     PC,POSHDS       ;POSITION HEADS
5542  026204  026346                            #T2365$
5543  026206                            BGNSUB
 (3)  026206                                                                           T7.1:
 (3)  026206  104002                        EMT     C$BSUB
5544  026210  012737  000400  002514            MOV     #256.,NEWCYL    ;SET FOR INNER GUARD BAND SEEK
5545  026216  004737  015624                    JSR     PC,XSEEK        ;DO IT
5546  026222  026322                            #60$
5547  026224  012701  000003                    MOV     #3.,R1          ;SET WAIT COUNT 3 MS
5548
5549  026230  032762  000001  000000  7$:       BIT     #DRDYMSK,RLCS(R2) ;CHECK IF READY
5550  026236  001413                            BEQ     9$              ;NO-SKIP
5551  026240  004737  014762                    JSR     PC,GSTAT        ;GET DRIVE STATUS
5552  026244  026322                            60$
5553  026246  012703  007333                    MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
5554  026252  012704  010246                    MOV     #C1OMS,R4       ;SET CONDITION MESSAGE PTR
5555  026256                                    EPRHRD  701.,,ERR4      ;REPORT READY ERROR
 (3)  026256  104443                        TRAP    T$ERCODE
 (5)  026260  001275                        .WORD   701
 (5)  026262  011526                        .WORD   ERR4
5556  026264  000416                        BR      60$             ;EXIT TEST
5557  026266  005301                9$:     DEC     R1              ;DEC WAIT COUNT
5558  026270  001404                        BEQ     11$             ;SKIP IF 0
5559  026272                                WAITUS  #10.            ;WAIT 100 US
 (3)  026272  012700  000012                MOV     #10.,R0
 (3)  026276  104027                        EMT     C$WTU
5560  026300  000753                        BR      7$              ;LOOP
5561  026302  012701  000226        11$:    MOV     #150.,R1                ;SET WAIT COUNT 15 MS
5562  026306  004737  020404                JSR     PC,RDYWAIT      ;GO WAIT FOR READY
5563  026312  026322                        #60$
5564
5565  026314  004737  021000                JSR     PC,VERPOS       ;GO VERIFY POSITION IS 255
5566  026320  026322                        #60$
5567  026322  012737  000002  002430  60$:   MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5568  026330                            ENDSUB
 (3)  026330                            L10035:
 (3)  026330  104003                        EMT     C$ESUB
5569  026332                                ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  026332  104010                        EMT     C$ESCAPE
 (3)  026334  000012                        .WORD   L10034-.
5570  026336  004737  017150                JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
5571  026342  026346                        15$                     ;ABORT RETURN
5572  026344  000713                        BR      T233$           ;REPEAT THE TESTS
5573  026346                            15$:
5574  026346                            T2365$:
```

```
 5575  026346                          ENDTST
   (3) 026346                          L10034:
   (3) 026346  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 8         **INCREMENTAL REVERSE SEEK HEAD 1
                                        BGNTST                  ;TEST 8
                                                                                                T8::
                                        MOV     #P2T08E,ERHEAD  ;SET ERROR HEADER
                                        JSR     PC,TSTINT       ;INITIALIZE TEST
                                        JSR     PC,GSTATR       ;GET STATUS & CLEAR
                                        #T2465$
                                        CLR     DESHD           ;SET TO HEAD 0
                                        MOV     HILIMW,R5       ;SET TO POSITION HDS AT HILIMIT
                                        JSR     PC,POSHDS       ;POSITION HDS
                                        #T2465$
                                        MOV     #1,DESHD        ;SET TO SELECT HD 1
                                        BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
                                        BEQ     2$              ;NO - SKIP
                                        TST     HEADW           ;TEST IF HEAD SPECIFIED IS 0
                                        BNE     2$              ;NO - SKIP
                                        EXIT    TST             ;ESLE EXIT TEST
                                        EMT     C$EXIT
                                        .WORD   L10036-.
                                2$:
                                BGNSUB
                                                                                                T8.1:
                                        EMT     C$BSUB
                                T247$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
                                        #60$
                                        INLOOP                  ;CHECK IF IN ERROR LOOP
                                        EMT     C$INLP
                                        BNCOMPLETE      5$      ;NO - SKIP
                                        BCC     5$
                                        CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
                                        BNE     5$              ;NO - SKIP
                                        JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDER
                                        BR      7$              ;SKIP
                                5$:     MOV     CURCYL,NEWCYL   ;MOV CUR TO NEW
                                        DEC     NEWCYL          ;DEC NEWCYL FOR 1 CYL REV SEEK
                                7$:     JSR     PC,XSEEK        ;DO SEEK
                                        #60$
                                        MOV     #150.,R1                ;SET WAIT FOR 15 MS
                                        JSR     PC,RDYWAIT      ;WAIT FOR READY
                                        #60$
                                        JSR     PC,VERPOS       ;VERIFY POSITION
                                        #60$
                                        BIT     #ALLSEC,MISWIW  ;TEST IF ALL SECTORS
                                        BEQ     9$              ;NO-EXIT
                                        JSR     PC,RDALHD       ;READ ALL HEADERS
                                        #60$
                                        JSR     PC,VERHDR       ;VERIFY HEADER
                                        #60$
                                9$:
                                        CMP     LOLIMW,NEWCYL   ;CHECK IF AT LOLIMIT
                                        BLO     T247$           ;NO - LOOP
                                60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
                                ENDSUB
                                L10037:
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:30  PAGE 84-15
DZRLDA.P11    04-NOV-77 11:51          *TEST 8          **INCREMENTAL REVERSE SEEK HEAD 1                    SEQ 0100

   (3)  026572  104003                          EMT     C$ESUB
 5625   026574                        T2465$:
 5626   026574                        ENDTST
   (3)  026574                        L10036:
   (3)  026574  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 9          **SEEK TESTS
5628
5629    026576                          BGNTST                   ;TEST 9
 (3)    026576                                                                  T9::
5630    026576  012737  006214  002424          MOV     #P2T09E,ERHEAD  ;SET ERROR HEADER
5631    026604  004737  014714                  JSR     PC,TSTINT       ;INITIALIZE TEST
5632    026610  004737  014732                  JSR     PC,GSTATR       ;CLEAR DRIVE
5633    026614  027070                          T2565$
5634    026616  004737  017124                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
5635    026622  013705  013164                  MOV     LOLIMW,R5       ;SET TO POSTION HEADS TO LOLIMIT
5636    026626  004737  016324                  JSR     PC,POSHDS       ;POSITION HDS TO LOWLIMIT
5637    026632  027070                          T2565$
5638    026634  004737  020652          T256$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
5639    026640  027070                          T2565$
5640    026642  013737  002516  002514          MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
5641    026650  012704  002274                  MOV     #T25TBL,R4      ;SET POINTER TO TABLE OF SEEK DIFF
5642    026654  012405          T258$:  MOV     (R4)+,R5        ;PUT FIRST IN R5
5643    026656  013701  013166                  MOV     HILIMW,R1       ;GET HILIMIT
5644    026662  163701  013164                  SUB     LOLIMW,R1       ;SUBTRACT LOLIMIT
5645    026666  021401                          CMP     (R4),R1         ;CHECK IF NEW DIFFERENCE IS IN BOUNDS
5646    026670  101073                          BHI     T2517$$         ;NO - SKIP TEST
5647    026672  060537  002514          T257$:  ADD     R5,NEWCYL       ;ADD TO PRESENT POSITION
5648    026676  023737  002514  013164          CMP     NEWCYL,LOLIMW   ;CHECK IF AT OR PAST LOLIMIT
5649    026704  002004                          BGE     9$              ;NO - SKIP
5650    026706  013737  013164  002514          MOV     LOLIMW,NEWCYL   ;ELSE SET TO LOLIMIT
5651    026714  000407                          BR      11$
5652    026716  023737  002514  013166  9$:     CMP     NEWCYL,HILIMW   ;CHECK IF AT HILIMIT OR GREATER
5653    026724  003403                          BLE     11$             ;NO - SKIP
5654    026726  013737  013166  002514          MOV     HILIMW,NEWCYL   ;ELSE SET FOR HILIMIT
5655    026734                          11$:
5656    026734                          BGNSUB
 (3)    026734                                                                  T9.1:
 (3)    026734  104002                          EMT     C$BSUB
5657    026736                          INLOOP                   ;CHECK IF IN ERROR LOOP
 (3)    026736  104020                  EMT     C$INLP
5658    026740                          BNCOMPLETE      13$      ;NO - SKIP
 (2)    026740  103011                  BCC     13$
5659    026742  004737  020652          JSR     PC,GETPOS       ;GET CURRENT POSITION
5660    026746  027012                  60$
5661    026750  023737  002516  002514          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED POSITION
5662    026756  001002                          BNE     13$             ;NO - SKIP
5663    026760  004737  017210                  JSR     PC,ONSWAP       ;ELSE SWAP CURRENT AND NEW CYLINDERS
5664    026764  004737  015624          13$:    JSR     PC,XSEEK        ;DO SEEK
5665    026770  027012                          60$
5666    026772  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT
5667    026776  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR READY
5668    027002  027012                          60$
5669    027004  004737  021000                  JSR     PC,VERPOS       ;VERIFY POSITION
5670    027010  027012                          60$
5671    027012  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
5672    027020                          ENDSUB
5673    027020                          L10041:
 (3)    027020  104003                  EMT     C$ESUB
5673    027022                          ESCAPE  TST              ;EXIT TEST IF ERROR
 (3)    027022  104010                  EMT     C$ESCAPE
 (3)    027024  000044                  .WORD   L10040-.
5674    027026  023737  013166  002514          CMP     HILIMW,NEWCYL   ;CHECK IF SEEK WAS TO HILIMIT
```

```
 5675  027034  001002                          BNE     15$             ;NO - SKIP
 5676  027036  005405                          NEG     R5              ;ELSE SET R5 TO REPEAT DIFF IN REVERSE
 5677  027040  000714                          BR      T257$
 5678  027042  023737  013164  002514  15$:    CMP     LOLIMW,NEWCYL   ;TEST IF LAST SEEK WAS TO LOLIMIT
 5679  027050  001310                          BNE     T257$           ;NO - GO DO SEEK TEST
 5680  027052  021427  000377                  CMP     (R4),#255.      ;CHECK IF ALL TABLE DIFF USED
 5681  027056  001276                          BNE     T258$           ;NO - SKIP
 5682  027060  004737  017150          T2517$: JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 5683  027064  027070                          T2565$                  ;ABORT RETURN
 5684  027066  000662                          BR      T256$           ;REPEAT TEST HEAD 1
 5685  027070                          T2565$:
 5686  027070                          ENDTST
   (3) 027070                          L10040:
   (3) 027070  104001                          EMT     C$ETST
```

```
5688                                    .SBTTL  *TEST 10         **FORWARD OSCILLATING SEEK
5689   027072                           BGNTST          ;TEST 10
 (3)   027072                                                                    T10::
5690   027072  012737  006221  002424           MOV     #P2T10E,ERHEAD  ;SET ERROR HEADER
5691   027100  004737  014714                   JSR     PC,TSTINT       ;INITIALIZE TEST
5692   027104  004737  014732                   JSR     PC,GSTATR       ;CLEAR DRIVE
5693   027110  027366                           T2665$
5694   027112  004737  017124                   JSR     PC,CHOSHD       ;GO CHOSE HEAD
5695   027116  012705  000001           T266$:  MOV     #1,R5           ;LOAD R5 FOR FIRST SEEK
5696   027122  032737  020000  013162           BIT     #HICYL,MISWIW   ;TEST IF HI CYLINDER SPECED
5697   027130  001402                           BEQ     2$              ;NO - SKIP
5698   027132  013705  013166                   MOV     HILIMW,R5       ;ELSE SET UPPER LIMIT
5699   027136  005037  002514           2$:     CLR     NEWCYL          ;SET TO SEEK TO CYL 0
5700   027142  032737  040000  013162           BIT     #LOCYL,MISWIW   ;CHECK IF LO CYL SPEC'D
5701   027150  001403                           BEQ     5$              ;NO - SKIP
5702   027152  013737  013164  002514           MOV     LOLIMW,NEWCYL   ;ELSE SET LOWER LIMIT
5703   027160  004737  015624           5$:     JSR     PC,XSEEK        ;DO SEEK
5704   027164  027366                           T2665$
5705   027166  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 120 MS
5706   027172  004737  020404                   JSR     PC,RDYWAIT      ;WAIT FOR READY
5707   027176  027366                           T2665$
5708   027200  004737  020652           T267$:  JSR     PC,GETPOS       ;GET HEAD POSITION
5709   027204  027366                           T2665$
5710   027206  010537  002514                   MOV     R5,NEWCYL       ;LOAD NEW CYLINDER INTO NEWCYL
5711   027212                           BGNSUB
 (3)   027212                                                                   T10.1:
 (3)   027212  104002                           EMT     C$BSUB
5712   027214                                   INLOOP                  ;CHECK IF IN ERROR LOOP
 (3)   027214  104020                           EMT     C$INLP
5713   027216                                   BNCOMPLETE      18$     ;NO - SKIP
 (2)   027216  103011                           BCC     18$
5714   027220  004737  020652                   JSR     PC,GETPOS       ;GET POSITION
5715   027224  027322                           60$
5716   027226  023737  002516  002514           CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED LOC
5717   027234  001002                           BNE     18$             ;NO - SKIP
5718   027236  004737  017210                   JSR     PC,ONSWAP               ;SWAP OLD AND NEW
5719   027242  004737  015624           18$:    JSR     PC,XSEEK        ;DO SEEK
5720   027246  027322                           60$
5721   027250  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT 120 MS
5722   027254  004737  020404                   JSR     PC,RDYWAIT      ;WAIT FOR READY
5723   027260  027322                           60$
5724   027262  004737  021000                   JSR     PC,VERPOS       ;VERIFY HEAD POSITION
5725   027266  027322                           60$
5726   027270  005737  002522                   TST     DESSGN          ;TEST IF JUST SEEK REV
5727   027274  001412                           BEQ     60$             ;YES - SKIP
5728   027276  005037  002514                   CLR     NEWCYL          ;ELSE SET TO SEEK TO 0
5729   027302  032737  040000  013162           BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
5730   027310  001754                           BEQ     18$             ;NO - SKIP
5731   027312  013737  013164  002514           MOV     LOLIMW,NEWCYL   ;ELSE SET LOW LIMIT FOR SEEK
5732   027320  000750                           BR      18$
5733   027322  012737  000002  002430  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5734   027330                           ENDSUB
 (3)   027330                           L10043:
 (3)   027330  104003                           EMT     C$ESUB
5735   027332                                   ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)   027332  104010                           EMT     C$ESCAPE
```

```
  (3)  027334  000032                          .WORD   L10042-.
5736   027336  032737  020000  013162          BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC'D
5737   027344  001004                          BNE     20$             ;YES - SKIP
5738   027346  005205                          INC     R5              ;BUMP R5
5739   027350  020527  000400                  CMP     R5,#256.        ;ALL CYLINDERS DONE
5740   027354  001311                          BNE     T267$           ;NO - GO DO NEXT CYLINDER
5741   027356  004737  017150          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
5742   027362  027366                          T2665$                  ;ABORT RETURN
5743   027364  000654                          BR      T266$           ;GO DO TESTS
5744   027366                          T2665$:
5745   027366                          ENDTST
  (3)  027366                          L10042:
  (3)  027366  104001                          EMT     CSETST
```

```
                                        .SBTTL  *TEST 11         **REVERSE OSCILLATING SEEK
5748  027370                            BGNTST                   ;TEST 11
 (3)  027370                                                                        T11::
5749  027370  012737  006236  002424            MOV     #P2T11E,ERHEAD  ;SET ERROR HEADER
5750  027376  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
5751  027402  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
5752  027406  027664                            T2765$
5753  027410  004737  017124                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
5754  027414  012737  000377  002514  T275$:    MOV     #255.,NEWCYL    ;SEEK OUT TO 255.
5755  027422  032737  020000  013162            BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC'D
5756  027430  001403                            BEQ     2$              ;NO - SKIP
5757  027432  013737  013166  002514            MOV     HILIMW,NEWCYL   ;ELSE SET UPPER LIMIT
5758  027440  012705  000376            2$:     MOV     #254.,R5        ;SET R5 FOR FIRST SEEKS
5759  027444  032737  040000  013162            BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
5760  027452  001402                            BEQ     5$              ;NO - SKIP
5761  027454  013705  013164                    MOV     LOLIMW,R5       ;SET LOWER LIMIT
5762  027460  004737  015624            5$:     JSR     PC,XSEEK        ;DO SEEK
5763  027464  027664                            T2765$
5764  027466  012701  005670                    MOV     #3000.,R1       ;SET WAIT TO 120 MS
5765  027472  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
5766  027476  027664                            T2765$
5767  027500  004737  020652            T276$:  JSR     PC,GETPOS       ;GET POSITION
5768  027504  027664                            T2765$
5769  027506  010537  002514                    MOV     R5,NEWCYL       ;SET FOR NEXT SEEK
5770  027512                            BGNSUB
 (3)  027512                                                                        T11.1:
 (3)  027512  104002                            EMT     C$BSUB
5771  027514                                    INLOOP                  ;CHECK IF IN ERROR LOOP
 (3)  027514  104020                            EMT     C$INLP
5772  027516                                    BNCOMPLETE      18$     ;NO - SKIP
 (2)  027516  103011                            BCC     18$
5773  027520  004737  020652                    JSR     PC,GETPOS       ;ELSE GET POSITION
5774  027524  027624                            60$
5775  027526  023737  002516  002514            CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED CYL
5776  027534  001002                            BNE     18$             ;NO - SKIP
5777  027536  004737  017210                    JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
5778  027542  004737  015624            18$:    JSR     PC,XSEEK        ;DO SEEK
5779  027546  027624                            60$
5780  027550  012701  005670                    MOV     #3000.,R1       ;SET WAIT FOR 120 MS
5781  027554  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
5782  027560  027624                            60$
5783  027562  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
5784  027566  027624                            60$
5785  027570  005737  002522                    TST     DESSGN          ;CHECK IF JUST SEEK FWD
5786  027574  001013                            BNE     60$             ;YES - SKIP
5787  027576  012737  000377  002514            MOV     #255.,NEWCYL    ;ELSE SEEK TO TO 255
5788  027604  032737  020000  013162            BIT     #HICYL,MISWIW   ;TEST IF HILIMIT SPEC'D
5789  027612  001753                            BEQ     18$             ;NO - SKIP
5790  027614  013737  013166  002514            MOV     HILIMW,NEWCYL   ;SET TO UPPER LIMIT
5791  027622  000747                            BR      18$
5792  027624  012737  000002  002430  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5793  027632                            ENDSUB
 (3)  027632                            L10045:
 (3)  027632  104003                            EMT     C$ESUB
5794  027634                            ESCAPE  TST                     ;EXIT TEST IF ERROR
 (3)  027634  104010                            EMT     C$ESCAPE
```

```
  (3)  027636  000026                          .WORD   L10044-.
5795   027640  032737  040000  013162          BIT     #LOCYL,MISWIW   ;TEST IF LOLIMIT SPEC'D
5796   027646  001002                          BNE     20$             ;YES - SKIP
5797   027650  005305                          DEC     R5              ;DEC CYLINDER COUNT
5798   027652  100312                          BPL     T276$           ;IF STILL POSITIVE, DO SEEKS AGAIN
5799   027654  004737  017150          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
5800   027660  027664                          T2765$                  ;ABORT RETURN
5801   027662  000654                          BR      T275$           ;LOOP AGAIN
5802   027664                          T2765$:
5803   027664                          ENDTST
  (3)  027664                          L10044:
  (3)  027664  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 12         **SEEK TIMING
027666                                  BGNTST                   ;TEST 12
(3)  027666                                                                      T12::
5807 027666 012737 006253 002424        MOV     #P2T12E,ERHEAD   ;SET ERROR HEADER
5808 027674 005737 002652               TST     PASNUM           ;TEST IF PASS 0
5809 027700 001003                      BNE     2$               ;NO - SKIP
5810 027702 005737 013162               TST     MISWIW           ;TEST IF MANUAL TESTS WERE RUN
5811 027706 100402                      BMI     1$               ;YES - SKIP
5812 027710 000137 031432       2$:     JMP     65$              ;ELSE EXIT TEST
5813 027714 004737 014714       1$:     JSR     PC,TSTINT        ;INITIALIZE TEST
5814 027720 004737 014732               JSR     PC,GSTATR        ;CLEAR DRIVE
5815 027724 031432                      65$
5816 027726 012700 002552               MOV     #OFIN,R0         ;GET ADDRESS OF 1ST TIME VALUE
5817 027732 012701 000030               MOV     #24.,R1          ;SET COUNT FOR CLEAR
5818 027736 005020              4$:     CLR     (R0)+            ;CLEAR TIMER STORAGE
5819 027740 005301                      DEC     R1
5820 027742 001375                      BNE     4$
5821 027744 005037 002644               CLR     PASCNT           ;CLEAR PASS COUNTER
5822 027750 005037 002514               CLR     NEWCYL           ;POSITION HEADS AT 0
5823 027754 004737 015624               JSR     PC,XSEEK         ;DO SEEK
5824 027760 031432                      65$
5825 027762 012701 005670               MOV     #3000.,R1        ;SET WAIT FOR 300 MS
5826 027766 004737 020404               JSR     PC,RDYWAIT       ;WAIT FOR READY
5827 027772 031432                      65$
5828 027774 004737 021000               JSR     PC,VERPOS        ;VERIFY POSITION
5829 030000 031432                      65$
5830 030002 004737 017124               JSR     PC,CHOSHD        ;GO CHOSE HEAD
5831 030006 012700 002562               MOV     #OFOUT,R0        ;SET PTRS FOR 1 CYL FWD OUTER TIMER
5832 030012 012701 002564               MOV     #OFOUTU,R1
5833 030016 012703 002576               MOV     #OROUT,R3
5834 030022 012704 002600               MOV     #OROUTU,R4
5835 030026 012737 000001 002514        MOV     #1,NEWCYL        ;SET NEWCYL TO CYL 1
5836 030034 012737 000200 002646 8$:    MOV     #128.,COUNT      ;SET COUNTER FOR SEEK LOOP
5837 030042 012737 000110 002550        MOV     #RDHEAD,TEMP8    ;BUILD READ HEADER COMMAND
5838 030050 053737 002444 002550        BIS     RLDRV,TEMP8
5839 030056 042737 002000 002550        BIC     #BIT10,TEMP8
5840 030064 004737 015614       9$:     JSR     PC,XSEEKT        ;DO SEEK BUILD BUT DO NOT START
5841 030070 031432                      65$
5842 030072 013762 002452 000004        MOV     L.DA,RLDA(R2)    ;LOAD RL REGISTERS
5843 030100 013762 002446 000000        MOV     L.CS,RLCS(R2)
5844 030106 010046                      MOV     R0,-(SP)         ;STORE R0
5845 030110                             WAITUS  #10.             ;WAIT FOR INTERRUPT
(3)  030110 012700 000012               MOV     #10.,R0
(3)  030114 104027                      EMT     C$WTU
5846 030116 005737 002420               TST     DONE             ;TEST IF INTERRUPT
5847 030122 001010                      BNE     17$              ;YES - SKIP
5848 030124 004737 014562               JSR     PC,WAITIN        ;WAIT FOR INTERRUPT
5849 030130 012603                      MOV     (SP)+,R3         ;GET MESSAGE POINTER
5850 030132                             ERRHRD  1201.,,ERR1
(3)  030132 104443                      TRAP    T$ERCODE
(5)  030134 002261                      .WORD   1201
(5)  030136 011344                      .WORD   ERR1
5851 030140 000137 031432               JMP     65$
5852 030144 005737 002456       17$:    TST     T.CS             ;CHECK IF ANY ERRORS
5853 030150 100005                      BPL     14$              ;NO - SKIP
5854 030152                             ERRHRD  1202.,,ERR6
```

```
   (3)  030152  104443                        TRAP    T$ERCODE
   (5)  030154  002262                        .WORD   1202
   (5)  030156  011646                        .WORD   ERR6
  5855  030160  000137  031432                JMP     65$
  5856  030164  005037  002420          14$:  CLR     DONE              ;CLEAR INTERRUPT FLAG
  5857  030170  013762  002550  000000      MOV     TEMP8,RLCS(R2)    ;LOAD RL REGISTER
  5858  030176                              WAITUS  #2000.            ;WAIT FOR INTERRUPT
   (3)  030176  012700  003720              MOV     #2000.,R0
   (3)  030202  104027                      EMT     C$WTU
  5859  030204                              GETTIM  R5                ;GET TIME USED
   (3)  030204  104052                      EMT     C$GTIM
   (3)  030206  010005                      MOV     R0,R5
  5860  030210  012600                      MOV     (SP)+,R0          ;RESTORE R0
  5861  030212  013737  002550  002446      MOV     TEMP8,L.CS        ;SET IF ERROR TO REPORT
  5862  030220  004737  021000              JSR     PC,VERPOS         ;VERIFY POSITION
  5863  030224  031432                      65$
  5864  030226  005737  002522              TST     DESSGN            ;CHECK WHICH SEEK DIRECTION
  5865  030232  001403                      BEQ     15$               ;REVERSE - SKIP
  5866  030234  060510                      ADD     R5,(R0)           ;ADD TO FORWARD TOTAL
  5867  030236  005511                      ADC     (R1)              ;ADD IN OVERFLOW
  5868  030240  000402                      BR      16$               ;SKIP
  5869  030242  060513          15$:  ADD     R5,(R3)           ;ADD TO REVERSE TOTAL
  5870  030244  005514                      ADC     (R4)              ;ADD IN OVERFLOW
  5871  030246  005337  002646          16$:  DEC     COUNT             ;DEC SEEK COUNT
  5872  030252  001403                      BEQ     18$               ;SKIP IF 0
  5873  030254  004737  017210              JSR     PC,ONSWAP         ;ELSE SWAP OLD AND NEW CYL
  5874  030260  000701                      BR      9$                ;REDO SEEK LOOP
  5875  030262  162710  000470          18$:  SUB     #312.,(R0)        ;SUB CONSTANT FOR READ HEADER TIME
  5876  030266  162713  000470                  SUB     #312.,(R3)
  5877  030272  012705  000006                  MOV     #6,R5             ;SET SHIFT COUNT TO DIVIDE BY 64
  5878  030276  000241          10$:  CLC                       ;DIVIDE BOTH TOTALS BY 64
  5879  030300  006011                      ROR     (R1)
  5880  030302  006010                      ROR     (R0)
  5881  030304  000241                      CLC
  5882  030306  006014                      ROR     (R4)
  5883  030310  006013                      ROR     (R3)
  5884  030312  005305                      DEC     R5
  5885  030314  001370                      BNE     10$
  5886  030316  005237  002644              INC     PASCNT            ;BUMP PASS COUNT
  5887  030322  022737  000001  002644      CMP     #1,PASCNT         ;TEST IF PASS 1
  5888  030330  001033                      BNE     24$               ;NO - SKIP
  5889  030332  012737  000177  002514      MOV     #127.,NEWCYL      ;ELSE SET TO POSITION HDS TO 127
  5890  030340  004737  015624              JSR     PC,XSEEK          ;DO SEEK
  5891  030344  031432                      65$
  5892  030346  012701  005670              MOV     #3000.,R1         ;SET WAIT COUNT FOR 300 MS
  5893  030352  004737  020404              JSR     PC,RDYWAIT        ;WAIT FOR READY
  5894  030356  031432                      65$
  5895  030360  004737  021000              JSR     PC,VERPOS         ;VERIFY POSITION
  5896  030364  031432                      65$
  5897  030366  012700  002556              MOV     #OFMID,R0         ;SET PTRS FOR TIMING 1 CYL SK AT 127
  5898  030372  012701  002560              MOV     #OFMIDU,R1
  5899  030376  012703  002572              MOV     #ORMID,R3
  5900  030402  012704  002574              MOV     #ORMIDU,R4
  5901  030406  012737  000200  002514      MOV     #128.,NEWCYL      ;SET NEWCYL TO 128
  5902  030414  000137  030034              JMP     8$                ;DO SEEK LOOP
  5903  030420  022737  000002  002644  24$:  CMP     #2,PASCNT         ;TEST IF PASS 2
```

```
5904  030426  001033                          BNE     28$             ;NO - SKIP
5905  030430  012737  000376  002514          MOV     #254.,NEWCYL    ;SET UP TO TIME 1 CYL SEEK AT INNER
5906  030436  004737  015624                  JSR     PC,XSEEK        ;  LIMIT
5907  030442  031432                          65$
5908  030444  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
5909  030450  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR READY
5910  030454  031432                          65$
5911  030456  004737  021000                  JSR     PC,VERPOS       ;VERIFY POSITION
5912  030462  031432                          65$
5913  030464  012700  002552                  MOV     #OFIN,R0        ;SET POINTERS
5914  030470  012701  002554                  MOV     #OFINU,R1
5915  030474  012703  002566                  MOV     #ORIN,R3
5916  030500  012704  002570                  MOV     #ORINU,R4
5917  030504  012737  000377  002514          MOV     #255.,NEWCYL    ;LOAD NEW CYLINDER
5918  030512  000137  030034                  JMP     8$              ;DO SEEK LOOP
5919  030516  022737  000003  002644  28$:    CMP     #3,PASCNT       ;TEST IF PASS 3
5920  030524  001031                          BNE     32$             ;NO - SKIP
5921  030526  005037  002514                  CLR     NEWCYL          ;ELSE SET UP TO TIME 128 CYL SEEK
5922  030532  004737  015624                  JSR     PC,XSEEK        ;  AT OUTER LIMIT
5923  030536  031432                          65$
5924  030540  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
5925  030544  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
5926  030550  031432                          65$
5927  030552  004737  021000                  JSR     PC,VERPOS       ;VERIFY POSITION
5928  030556  031432                          65$
5929  030560  012700  002606                  MOV     #HFOUT,R0       ;SET POINTERS
5930  030564  012701  002610                  MOV     #HFOUTU,R1
5931  030570  012703  002616                  MOV     #HROUT,R3
5932  030574  012704  002610                  MOV     #HFOUTU,R4
5933  030600  012737  000177  002514          MOV     #127.,NEWCYL    ;LOAD NEWCYL FOR 128 CYL SEEK
5934  030606  000472                          BR      39$
5935  030610  022737  000004  002644  32$:    CMP     #4,PASCNT       ;TEST IF PASS 4
5936  030616  001032                          BNE     36$             ;NO - SKIP
5937  030620  012737  000200  002514          MOV     #128.,NEWCYL    ;ELSE SET UP TO TIME 128 CYL SEEK
5938  030626  004737  015624                  JSR     PC,XSEEK        ;  AT INNER LIMIT
5939  030632  031432                          65$
5940  030634  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
5941  030640  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR READY
5942  030644  031432                          65$
5943  030646  004737  021000                  JSR     PC,VERPOS       ;VERIFY POSITION
5944  030652  031432                          65$
5945  030654  012700  002602                  MOV     #HFIN,R0        ;SET POINTERS
5946  030660  012701  002604                  MOV     #HFINU,R1
5947  030664  012703  002612                  MOV     #HRIN,R3
5948  030670  012704  002614                  MOV     #HRINU,R4
5949  030674  012737  000377  002514          MOV     #255.,NEWCYL    ;SET NEWCYL TO 255 FOR 128 CYL SEEK
5950  030702  000434                          BR      39$             ;DO TIMING LOOP
5951  030704  022737  000005  002644  36$:    CMP     #5,PASCNT       ;TEST IF PASS 5
5952  030712  001032                          BNE     40$             ;NO - SKIP
5953  030714  005037  002514                  CLR     NEWCYL          ;ELSE SET UP TO TIME 256 CYL SEEK
5954  030720  004737  015624                  JSR     PC,XSEEK        ;  OVER ALL SURFACE
5955  030724  031432                          65$
5956  030726  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
5957  030732  004737  020404                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
5958  030736  031432                          65$
5959  030740  004737  021000                  JSR     PC,VERPOS       ;VERIFY POSITION
```

```
 5960  030744  031432                          65$
 5961  030746  012700  002622                  MOV     #AFMID,R0       ;SET POINTERS
 5962  030752  012701  002624                  MOV     #AFMIDU,R1
 5963  030756  012703  002626                  MOV     #ARMID,R3
 5964  030762  012704  002630                  MOV     #ARMIDU,R4
 5965  030766  012737  000377  002514          MOV     #255.,NEWCYL    ;SET NEWCYL
 5966  030774  000137  030034          39$:    JMP     8$
 5967  031000                          40$:    PRINTF  #FMT1.1,#SKTMES,#VALDES
  (9)  031000  012746  006547                  MOV     #VALDES,-(SP)
  (8)  031004  012746  006504                  MOV     #SKTMES,-(SP)
  (7)  031010  012746  010421                  MOV     #FMT1.1,-(SP)
  (6)  031014  012746  000003                  MOV     #3,-(SP)
  (3)  031020  010600                          MOV     SP,R0
  (4)  031022  104017                          EMT     C$PNTF
  (4)  031024  062706  000010                  ADD     #10,SP
 5968  031030                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  031030  005046                          CLR     -(SP)
 (11)  031032  153716  002445                  BISB    RLDRV+1,(SP)
 (10)  031036  012746  005423                  MOV     #DRVNAM,-(SP)
  (9)  031042  013746  002440                  MOV     RLBAS,-(SP)
  (8)  031046  012746  005412                  MOV     #BASADD,-(SP)
  (7)  031052  012746  010447                  MOV     #FMT5,-(SP)
  (6)  031056  012746  000005                  MOV     #5,-(SP)
  (3)  031062  010600                          MOV     SP,R0
  (4)  031064  104017                          EMT     C$PNTF
  (4)  031066  062706  000014                  ADD     #14,SP
 5969  031072                                  PRINTF  #FMT18,#LABIN,#LABMID,#LABOUT,#LABEXP
 (11)  031072  012746  006641                  MOV     #LABEXP,-(SP)
 (10)  031076  012746  006633                  MOV     #LABOUT,-(SP)
  (9)  031102  012746  006624                  MOV     #LABMID,-(SP)
  (8)  031106  012746  006616                  MOV     #LABIN,-(SP)
  (7)  031112  012746  011041                  MOV     #FMT18,-(SP)
  (6)  031116  012746  000005                  MOV     #5,-(SP)
  (3)  031122  010600                          MOV     SP,R0
  (4)  031124  104017                          EMT     C$PNTF
  (4)  031126  062706  000014                  ADD     #14,SP
 5970  031132                                  PRINTF  #FMT19,#LABOCF,OFIN,OFMID,OFOUT,EXOCYL
 (12)  031132  013746  002632                  MOV     EXOCYL,-(SP)
 (11)  031136  013746  002562                  MOV     OFOUT,-(SP)
 (10)  031142  013746  002556                  MOV     OFMID,-(SP)
  (9)  031146  013746  002552                  MOV     OFIN,-(SP)
  (8)  031152  012746  006652                  MOV     #LABOCF,-(SP)
  (7)  031156  012746  011073                  MOV     #FMT19,-(SP)
  (6)  031162  012746  000006                  MOV     #6,-(SP)
  (3)  031166  010600                          MOV     SP,R0
  (4)  031170  104017                          EMT     C$PNTF
  (4)  031172  062706  000016                  ADD     #16,SP
 5971  031176                                  PRINTF  #FMT19,#LABOCR,ORIN,ORMID,OROUT,EXOCYL
 (12)  031176  013746  002632                  MOV     EXOCYL,-(SP)
 (11)  031202  013746  002576                  MOV     OROUT,-(SP)
 (10)  031206  013746  002572                  MOV     ORMID,-(SP)
  (9)  031212  013746  002566                  MOV     ORIN,-(SP)
  (8)  031216  012746  006666                  MOV     #LABOCR,-(SP)
  (7)  031222  012746  011073                  MOV     #FMT19,-(SP)
  (6)  031226  012746  000006                  MOV     #6,-(SP)
  (3)  031232  010600                          MOV     SP,R0
```

```
  (4)  031234  104017                    EMT     C$PNTF
  (4)  031236  062706  000016.           ADD     #16,SP
 5972  031242                            PRINTF  #FMT20,#LABHCF,HFIN,HFOUT,EXHCYL
  (11) 031242  013746  002634            MOV     EXHCYL,-(SP)
  (10) 031246  013746  002606            MOV     HFOUT,-(SP)
  (9)  031252  013746  002602            MOV     HFIN,-(SP)
  (8)  031256  012746  006702            MOV     #LABHCF,-(SP)
  (7)  031262  012746  011130            MOV     #FMT20,-(SP)
  (6)  031266  012746  000005            MOV     #5,-(SP)
  (3)  031272  010600                    MOV     SP,R0
  (4)  031274  104017                    EMT     C$PNTF
  (4)  031276  062706  000014            ADD     #14,SP
 5973  031302                            PRINTF  #FMT20,#LABHCR,HRIN,HROUT,EXHCYL
  (11) 031302  013746  002634            MOV     EXHCYL,-(SP)
  (10) 031306  013746  002616            MOV     HROUT,-(SP)
  (9)  031312  013746  002612            MOV     HRIN,-(SP)
  (8)  031316  012746  006716            MOV     #LABHCR,-(SP)
  (7)  031322  012746  011130            MOV     #FMT20,-(SP)
  (6)  031326  012746  000005            MOV     #5,-(SP)
  (3)  031332  010600                    MOV     SP,R0
  (4)  031334  104017                    EMT     C$PNTF
  (4)  031336  062706  000014_           ADD     #14,SP
 5974  031342                            PRINTF  #FMT21,#LABACF,AFMID,EXACYL
  (10) 031342  013746  002636            MOV     EXACYL,-(SP)
  (9)  031346  013746  002622            MOV     AFMID,-(SP)
  (8)  031352  012746  006732            MOV     #LABACF,-(SP)
  (7)  031356  012746  011160            MOV     #FMT21,-(SP)
  (6)  031362  012746  000004            MOV     #4,-(SP)
  (3)  031366  010600                    MOV     SP,R0
  (4)  031370  104017                    EMT     C$PNTF
  (4)  031372  062706  000012            ADD     #12,SP
 5975  031376                            PRINTF  #FMT21,#LABACR,ARMID,EXACYL
  (10) 031376  013746  002636            MOV     EXACYL,-(SP)
  (9)  031402  013746  002626            MOV     ARMID,-(SP)
  (8)  031406  012746  006746            MOV     #LABACR,-(SP)
  (7)  031412  012746  011160            MOV     #FMT21,-(SP)
  (6)  031416  012746  000004            MOV     #4,-(SP)
  (3)  031422  010600                    MOV     SP,R0
  (4)  031424  104017                    EMT     C$PNTF
  (4)  031426  062706  000012            ADD     #12,SP
 5976  031432                        65$:
 5977  031432                        ENDTST
  (3)  031432                        L10046:
  (3)  031432  104001                    EMT     C$ETST
```

```
5979                                    .SBTTL  *TEST 13         **BASIC READ DATA (BAD SECTOR FILE)
5980   031434                           BGNTST                   ;TEST 13
 (3)   031434                                                                              T13::
5981   031434  012737  006267  002424            MOV     #P2T13E,ERHEAD  ;SET ERROR HEADER
5982   031442  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
5983   031446  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
5984   031452  032114                            65$
5985   031454  012737  000001  002524            MOV     #1,DESHD        ;SET TO HEAD 1
5986   031462  032737  010000  013162            BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPEC'D
5987   031470  001405                            BEQ     2$              ;NO - SKIP
5988   031472  005737  013170                    TST     HEADW           ;TEST IF HEAD 0
5989   031476  001002                            BNE     2$              ;NO - SKIP
5990   031500                           EXIT    TST                     ;ELSE EXIT TEST
 (3)   031500  104032                            EMT     C$EXIT
 (3)   031502  000440                            .WORD   L10047-.
5991   031504  012737  000377  002514  2$:       MOV     #255.,NEWCYL    ;POSITION HEADS AT 255
5992   031512  004737  015624                    JSR     PC,XSEEK        ;DO SEEK
5993   031516  032114                            65$
5994   031520  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
5995   031524  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR INTERRUPT
5996   031530  032114                            65$
5997   031532  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
5998   031536  032114                            65$
5999   031540  005037  002526                    CLR     DESSEC          ;SET FOR SECTOR 0
6000   031544  012737  003062  002542            MOV     #FBSFIL,TEMP5   ;SET TEMP STORAGE FOR FACTORY BS FILE
6001   031552  012737  000020  002544            MOV     #16.,TEMP6      ;SET MAX SECTOR COUNT
6002   031560  112737  000001  002657            MOVB    #1,NOERCT       ;SET FOR NO ERROR COUNTING
6003   031566  105037  002656                    CLRB    LOCERR          ;CLEAR LOCAL ERROR COUNTER
6004   031572  005037  002536           4$:      CLR     TEMP3           ;CLEAR ONES DETECTED FLAG
6005   031576  013701  002542                    MOV     TEMP5,R1        ;INIT POINTERS
6006   031602  013700  002544                    MOV     TEMP6,R0
6007   031606  012703  003256                    MOV     #IBUFF,R3
6008   031612  012737  000002  002430            MOV     #2,ERRSWI       ;INIT ERROR SWITCH
6009   031620  004737  022150                    JSR     PC,XREAD        ;DO READ
6010   031624  031776                            39$
6011   031626  005723                            TST     (R3)+           ;TEST IF WORD 0 NOT NEG
6012   031630  100515                            BMI     45$             ;YES, BAD FMT ERROR
6013   031632  005723                            TST     (R3)+           ;ELSE TEST WORD 1 NOT NEG
6014   031634  100513                            BMI     45$             ;YES - BAD FMT ERROR REPORT
6015   031636  005723                   7$:      TST     (R3)+           ;TEST WORD 2 IS 0
6016   031640  001111                            BNE     45$             ;NO - SKIP TO FMT ERROR RPT
6017   031642  005723                            TST     (R3)+           ;TEST WORD 3 IS 0
6018   031644  001107                            BNE     45$             ;NO - SKIP TO FMT ERROR RPT
6019   031646  021327  177777           8$:      CMP     (R3),#-1        ;TEST IF NEXT WORD IS ALL 1'S
6020   031652  001004                            BNE     10$             ;NO  - SKIP
6021   031654  012737  000001  002536            MOV     #1,TEMP3        ;ELSE SET 1'S DETECTED FLAG
6022   031662  000403                            BR      11$             ;SKIP
6023   031664  005737  002536           10$:     TST     TEMP3           ;TEST IF ONES HAVE BEEN DETECTED
6024   031670  001075                            BNE     45$             ;YES - SKIP TO FMT ERROR RPT
6025   031672  012311                   11$:     MOV     (R3)+,(R1)      ;STORE CYLINDER WORD
6026   031674  012705  000007                    MOV     #7,R5           ;ALIGN IT TO LOOK LIKE HEADER
6027   031700  006311                   12$:     ASL     (R1)
6028   031702  005305                            DEC     R5
6029   031704  001375                            BNE     12$
6030   031706  032713  000400                    BIT     #BIT8,(R3)      ;TEST IF HEAD 1
6031   031712  001402                            BEQ     15$             ;NO - SKIP
```

```
 6032  031714  052711  000100                 BIS     #BIT6,(R1)          ;INSERT HEAD BIT
 6033  031720  042713  177400          15$:   BIC     #177400,(R3)        ;CLEAR ALL BUT SECTOR
 6034  031724  052321                         BIS     (R3)+,(R1)+         ;INSERT SECTOR NUMBER
 6035  031726  020327  003656                 CMP     R3,#IBUFF+256.      ;CHECK IF IBUFF EMPTY
 6036  031732  001345                         BNE     8$                  ;NO GET NEXT CYLINDER
 6037  031734  005737  002536                 TST     TEMP3               ;ELSE TEST IF 1'S DETECTED
 6038  031740  001457                         BEQ     48$                 ;TO MANY ERRORS - REPORT
 6039  031742  022737  000044  002544         CMP     #36.,TEMP6          ;CHECK IF SOFTWARE BAD READ
 6040  031750  001461                         BEQ     65$                 ;YES - SKIP
 6041  031752  012737  002666  002542  37$:   MOV     #SBSFIL,TEMP5       ;ELSE CHANGE POINTERS
 6042  031760  012737  000044  002544         MOV     #36.,TEMP6          ;       MAX SECTOR NUMBER
 6043  031766  012737  000024  002526         MOV     #20.,DESSEC         ;       SECTOR NUMBER START
 6044  031774  000676                         BR      4$                  ;DO READ
 6045  031776  005237  002656          39$:   INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
 6046  032002  012777  177777  150532  40$:   MOV     #-1,@TEMP5          ;MOV    1'S INTO FILE STORAGE
 6047  032010                                 INLOOP                      ;CHECK IF IN ERROR LOOP
   (3) 032010  104020                         EMT     C$INLP
 6048  032012                                 BCOMPLETE       4$                  ;YES - GO DO READ
   (2) 032012  103667                         BCS     4$
 6049  032014  023737  002526  002544  41$:   CMP     DESSEC,TEMP6        ;CHECK IF ALL SECTORS READ
 6050  032022  001014                         BNE     43$                 ;NO - SKIP
 6051  032024  012703  005273                 MOV     #MBADSF,R3          ;SET RESULT MESSAGE POINTER
 6052  032030  005237  002656                 INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
 6053  032034                                 ERRHRD  1301.,,ERR1
   (3) 032034  104443                         TRAP    T$ERCODE
   (5) 032036  002425                         .WORD   1301
   (5) 032040  011344                         .WORD   ERR1
 6054  032042  022737  002666  002542         CMP     #SBSFIL,TEMP5       ;TEST IF SOFTWARE FILES CHECKED
 6055  032050  001421                         BEQ     65$                 ;YES - EXIT
 6056  032052  000737                         BR      37$                 ;ELSE GO CHECK SOFTWARE FILES
 6057  032054  062737  000004  002526  43$:   ADD     #4,DESSEC           ;BUMP TO NEXT SECTOR
 6058  032062  000643                         BR      4$                  ;GO DO READ
 6059  032064  012703  005323          45$:   MOV     #MFMTER,R3          ;SET RESULT MESSAGE POINTER
 6060  032070                                 ERRHRD  1302.,,ERR1
   (3) 032070  104443                         TRAP    T$ERCODE
   (5) 032072  002426                         .WORD   1302
   (5) 032074  011344                         .WORD   ERR1
 6061  032076  000737                         BR      39$                 ;GO CHECK FOR LOOP
 6062  032100  012703  005350          48$:   MOV     #MTMBS,R3           ;SET RESULT MESSAGE PTR
 6063  032104                                 ERRHRD  1303.,,ERR1
   (3) 032104  104443                         TRAP    T$ERCODE
   (5) 032106  002427                         .WORD   1303
   (5) 032110  011344                         .WORD   ERR1
 6064  032112  000733                         BR      40$                 ;GO CHECK FOR LOOP
 6065  032114  012737  000002  002430  65$:   MOV     #2,ERRSWI           ;INIT ERROR SWITCH
 6066  032122  012737  000001  002664         MOV     #1,BSFVAL           ;SET BAD SECTOR FILES VALID FLAG
 6067  032130  105737  002656                 TSTB    LOCERR              ;TEST IF LOCAL ERRORS
 6068  032134  001402                         BEQ     66$                 ;NO - SKIP
 6069  032136  005237  002650                 INC     ERRCNT              ;ELSE BUMP ERROR COUNT
 6070  032142                          66$:
 6071  032142                                 ENDTST
   (3) 032142                          L10047:
   (3) 032142  104001                         EMT     C$ETST
```

```
6073                                    .SBTTL  *TEST 14         **WRITE/READ DATA (PART 1)
6074  032144                            BGNTST          ;TEST 14
 (3)  032144                                                                              T14::
6075  032144  012737  006307  002424            MOV     #P2T14E,ERHEAD  ;SET ERROR HEADER
6076  032152  004737  017234                    JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
6077  032156  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
6078  032162  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
6079  032166  032356                            T3065$
6080  032170  004737  017124                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
6081  032174  005037  002526                    CLR     DESSEC          ;         SECTOR 0
6082  032200  005037  002514                    CLR     NEWCYL          ;         CYLINDER 0
6083  032204  005037  032250                    CLR     T310$           ;CLEAR PATTERN SELECT
6084  032210  004737  015624            T306$:  JSR     PC,XSEEK        ;POSITION HEADS
6085  032214  032356                            T3065$
6086  032216  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
6087  032222  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
6088  032226  032356                            T3065$
6089  032230  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
6090  032234  032356                            T3065$
6091  032236  005037  032250                    CLR     T310$           ;CLEAR PATTERN SELECTOR
6092  032242                            T307$:
6093  032242                            BGNSUB
                                                                                         T14.1:
 (3)  032242
 (3)  032242  104002                            EMT     C$BSUB
6094  032244  004537  021462                    JSR     R5,DATGEN       ;GENERATE DATA
6095  032250  000000                    T310$:  .WORD   0               ;PATTERN SELECT WORD
6096  032252  004737  022110                    JSR     PC,XWRITE       ;DO WRITE DATA
6097  032256  032274                            60$
6098  032260  004737  022150                    JSR     PC,XREAD        ;DO READ DATA
6099  032264  032274                            60$
6100  032266  004737  021622                    JSR     PC,DATCOM       ;COMPARE DATA
6101  032272  032274                            60$
6102  032274  012737  000002  002430    60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
6103  032302                            ENDSUB
 (3)  032302                            L10051:
 (3)  032302  104003                            EMT     C$ESUB
6104  032304                                    ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  032304  104010                            EMT     C$ESCAPE
 (3)  032306  000050                            .WORD   L10050-.
6105  032310  022737  000010  032250            CMP     #8.,T310$       ;WAS DATA PAT 8 USED?
6106  032316  001403                            BEQ     10$             ;YES - SKIP
6107  032320  005237  032250                    INC     T310$           ;ELSE BUMP TO NEXT PATTERN
6108  032324  000746                            BR      T307$           ;DO TEST WITH NEW PATTERN
6109  032326  004737  017150            10$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
6110  032332  032356                            T3065$                  ;ABORT RETURN
6111  032334  005037  032250                    CLR     T310$           ;SET PATTERN SELECT TO 0
6112  032340  004737  022630            11$:    JSR     PC,BSCHK        ;CHECK IF SECTOR BAD
6113  032344  032350                            13$                     ;YES RETURN - SKIP TO 13$
6114  032346  000720                            BR      T306$           ;NO RETURN - DO TEST THIS SECTOR
6115  032350  005237  002514            13$:    INC     NEWCYL          ;BUMP TO NEXT CYLINDER
6116  032354  000771                            BR      11$             ;CHECK IF THIS ONE BAD
6117  032356                            T3065$:
6118  032356                            ENDTST
 (3)  032356                            L10050:
 (3)  032356  104001                            EMT     C$ETST
6119
```

```
6121                                    .SBTTL  *TEST 15         **SPINDLE TIMING TEST
6122  032360                            BGNTST          ;TEST 15
 (3)  032360                                                                    T15::
6123  032360  012737  006332  002424            MOV     #P2T15E,ERHEAD  ;SET ERROR HEADER
6124  032366  005737  002652                    TST     PASNUM          ;TEST IF PASS 0
6125  032372  001003                            BNE     2$              ;NO - SKIP
6126  032374  005737  013162                    TST     MISWIW          ;TEST IF MANUAL TESTS WERE RUN
6127  032400  100402                            BMI     1$              ;YES - SKIP
6128  032402                            2$:     EXIT    TST             ;ELSE SKIP TEST
 (3)  032402  104032                            EMT     C$EXIT
 (3)  032404  000476                            .WORD   L10052-.
6129  032406  005003                    1$:     CLR     R3              ;CLEAR FOR TIMING STORAGE
6130  032410  005004                            CLR     R4
6131  032412  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
6132  032416  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
6133  032422  033074                            60$
6134  032424  004537  021462                    JSR     R5,DATGEN       ;GENERATE DATA
6135  032430  000000                            #0                      ;PATTERN 0
6136  032432  005037  002526                    CLR     DESSEC          ;CLEAR TO SECTOR 0
6137  032436  004737  017124                    JSR     PC,CHOSHD       ;GO SELECT HEAD
6138  032442  013737  013164  002514            MOV     LOLIMW,NEWCYL   ;SET FOR CYLINDER
6139  032450  004737  015624                    JSR     PC,XSEEK        ;DO SEEK
6140  032454  033074                            60$
6141  032456  012701  005670                    MOV     #3000.,R1       ;SET WAIT FOR 300 MS
6142  032462  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
6143  032466  033074                            60$
6144  032470  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
6145  032474  033074                            60$
6146  032476  012701  000100                    MOV     #64.,R1 ;SET LOOP COUNTER
6147  032502  012705  002454            5$:     MOV     #L.MP,R5        ;SET A POINTER
6148  032506  004737  022100                    JSR     PC,XWRITT       ;DO FIRST WRITE
6149  032512  033074                            60$
6150  032514  011562  000006                    MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS
6151  032520  014562  000004                    MOV     -(R5),RLDA(R2)
6152  032524  014562  000002                    MOV     -(R5),RLBA(R2)
6153  032530  014562  000000                    MOV     -(R5),RLCS(R2)
6154  032534                                    WAITUS  #3000.
 (3)  032534  012700  005670                    MOV     #3000.,R0
 (3)  032540  104027                            EMT     C$WTU
6155  032542  005737  002420                    TST     DONE            ;TEST IF INTERRUPT
6156  032546  001010                            BNE     6$              ;YES - SKIP
6157  032550  004737  014562                    JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
6158  032554  012603                            MOV     (SP)+,R3        ;GET MESSAGE POINTER
6159  032556                                    ERRHRD  1501.,,ERR1
 (3)  032556  104443                            TRAP    T$ERCODE
 (5)  032560  002735                            .WORD   1501
 (5)  032562  011344                            .WORD   ERR1
6160  032564  000137  033074                    JMP     60$
6161  032570  005737  002456            6$:     TST     T.CS            ;TEST IF ANY ERRORS
6162  032574  100005                            BPL     4$              ;NO - SKIP
6163  032576                                    ERRHRD  1502.,,ERR6
 (3)  032576  104443                            TRAP    T$ERCODE
 (5)  032600  002736                            .WORD   1502
 (5)  032602  011646                            .WORD   ERR6
6164  032604  000137  033074                    JMP     60$
6165  032610  012705  002454            4$:     MOV     #L.MP,R5        ;SET POINTER TO RL LOAD REGS
```

```
 6166  032614  005037  002420                CLR     DONE            ;CLEAR INTERRUPT INDICATOR
 6167  032620  011562  000006                MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS FOR 2ND WRITE
 6168  032624  014562  000004                MOV     -(R5),RLDA(R2)
 6169  032630  014562  000002                MOV     -(R5),RLBA(R2)
 6170  032634  014562  000000                MOV     -(R5),RLCS(R2)
 6171  032640                                WAITUS  #3000.          ;WAIT FOR INTERRUPT
  (3)  032640  012700  005670                MOV     #3000.,R0
  (3)  032644  104027                        EMT     C$WTU
 6172  032646                                GETTIM  R0              ;GET TIME WAITED
  (3)  032646  104052                        EMT     C$GTIM
 6173  032650  005737  002420                TST     DONE            ;TEST IN INTERRUPT OCCURRED
 6174  032654  001007                        BNE     7$              ;YES - SKIP
 6175  032656  004737  014562                JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
 6176  032662  012603                        MOV     (SP)+,R3        ;GET MESSAGE POINTER
 6177  032664                                ERRHRD  1503.,,ERR1     ;REPORT
  (3)  032664  104443                        TRAP    T$ERCODE
  (5)  032666  002737                        .WORD   1503
  (5)  032670  011344                        .WORD   ERR1
 6178  032672  000500                        BR      60$
 6179  032674  005737  002456          7$:   TST     T.CS            ;TEST IN ANY ERROR
 6180  032700  100004                        BPL     8$              ;NO - SKIP
 6181  032702                                ERRHRD  1504.,,ERR6     ;REPORT ERRORS
  (3)  032702  104443                        TRAP    T$ERCODE
  (5)  032704  002740                        .WORD   1504
  (5)  032706  011646                        .WORD   ERR6
 6182  032710  000471                        BR      60$
 6183  032712  060003          8$:   ADD     R0,R3           ;ADD IN TIME USED
 6184  032714  005504                        ADC     R4              ;DOUBLE PRECISSION
 6185  032716  005301                        DEC     R1              ;DEC LOOP COUNTER
 6186  032720  001270                        BNE     5$              ;LOOP UNTIL 0
 6187  032722  012701  000006                MOV     #6,R1           ;SET DIVIDE COUNT
 6188  032726  000241          10$:  CLC                             ;CLEAR CARRY FOR DIVIDE
 6189  032730  006004                        ROR     R4              ;DIVIDE SUM BY 100(8)
 6190  032732  006003                        ROR     R3
 6191  032734  005301                        DEC     R1              ;DEC DIVIDE COUNT
 6192  032736  001373                        BNE     10$             ;LOOP UNTIL DONE
 6193  032740                                PRINTF  #FMT1.1,#SRTMES,#VALDES
  (9)  032740  012746  006547                MOV     #VALDES,-(SP)
  (8)  032744  012746  006520                MOV     #SRTMES,-(SP)
  (7)  032750  012746  010421                MOV     #FMT1.1,-(SP)
  (6)  032754  012746  000003                MOV     #3,-(SP)
  (3)  032760  010600                        MOV     SP,R0
  (4)  032762  104017                        EMT     C$PNTF
  (4)  032764  062706  000010                ADD     #10,SP
 6194  032770                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  032770  005046                        CLR     -(SP)
 (11)  032772  153716  002445                BISB    RLDRV+1,(SP)
 (10)  032776  012746  005423                MOV     #DRVNAM,-(SP)
  (9)  033002  013746  002440                MOV     RLBAS,-(SP)
  (8)  033006  012746  005412                MOV     #BASADD,-(SP)
  (7)  033012  012746  010447                MOV     #FMT5,-(SP)
  (6)  033016  012746  000005                MOV     #5,-(SP)
  (3)  033022  010600                        MOV     SP,R0
  (4)  033024  104017                        EMT     C$PNTF
  (4)  033026  062706  000014                ADD     #14,SP
 6195  033032                                PRINTF  #FMT26,#RESE3,R3,#RESE4,#MAPROX,EXROT
```

```
   (12) 033032  013746  002640                    MOV     EXROT,-(SP)
   (11) 033036  012746  006606                    MOV     #MAPROX,-(SP)
   (10) 033042  012746  010177                    MOV     #RESE4,-(SP)
   (9)  033046  010346                            MOV     R3,-(SP)
   (8)  033050  012746  010173                    MOV     #RESE3,-(SP)
   (7)  033054  012746  011270                    MOV     #FMT26,-(SP)
   (6)  033060  012746  000006                    MOV     #6,-(SP)
   (3)  033064  010600                            MOV     SP,R0
   (4)  033066  104017                            EMT     C$PNTF
   (4)  033070  062706  000016                    ADD     #16,SP
  6196  033074  012737  000002  002430  60$:      MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
  6197  033102                        ENDTST
   (3)  033102                        L10052:
   (3)  033102  104001                            EMT     C$ETST
  6198
```

```
                                        .SBTTL  *TEST 16         **WRITE/READ DATA (PART 2)
033104                                  BGNTST                   ;TEST 16
(3) 033104                                                                              T16::
033104  012737  006362  002424                  MOV     #P2T16E,ERHEAD  ;SET ERROR HEADER
033112  004737  017234                          JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
033116  004737  014714                          JSR     PC,TSTINT       ;INITIALIZE TEST
033122  004737  014732                          JSR     PC,GSTATR       ;CLEAR DRIVE
033126  034124                                  T3165$
033130  005037  002644                          CLR     PASCNT          ;CLEAR PASS TO 0
033134  012705  177777                          MOV     #-1,R5          ;SET R5
033140  005737  002652                          TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
033144  001006                                  BNE     1$              ;NO - SKIP
033146  032737  000001  013162                  BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
033154  001002                                  BNE     1$              ;YES - SKIP
033156  012705  177770                          MOV     #-8.,R5         ;ELSE SET R5 TO NEG 8
033162                                  1$:
033162  012701  002322                          MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
033166  012703  000010                          MOV     #10,R3          ;SET CLEAR COUNT
033172  013721  013164                  2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LO LIMIT
033176  005303                                  DEC     R3              ;DEC COUNT
033200  001374                                  BNE     2$              ;LOOP UNTIL 0
033202  113737  013166  002326                  MOVB    HILIMW,T33TBL+4 ;INSERT HILIMIT
033210  113737  013166  002330                  MOVB    HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
033216  113737  013166  002332                  MOVB    HILIMW,T33TBL+10
033224  005205                          T3100$: INC     R5              ;BUMP R5
033226  032737  000001  013162                  BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
033234  001017                                  BNE     5$              ;YES - SKIP
033236  005737  002652                          TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
033242  001002                                  BNE     3$              ;NO - SKIP
033244  062705  000007                          ADD     #7,R5           ;ELSE BUMP CYLINDER POINTER BY 7
033250  020527  000051                  3$:     CMP     R5,#41.         ;TEST IF PAST TABLE
033254  103005                                  BHIS    4$              ;YES - GO TO EXIT

033256  116503  002342                          MOVB    CYLTBL(R5),R3   ;GET NEXT TABLE ENTRY
033262  042703  177400                          BIC     #177400,R3      ;CLEAR UPPER BYTE
033266  001006                                  BNE     8$              ;SKIP IF NOT 0
033270  000137  034124                  4$:     JMP     T3165$          ;EXIT TEST
033274  022705  000377                  5$:     CMP     #255.,R5        ;TEST IF ALL CYLINDERS USED
033300  001773                                  BEQ     4$              ;YES - EXIT TEST
033302  010503                                  MOV     R5,R3           ;USE R5 AS NEXT CYLINDER
033304  020337  013164                  8$:     CMP     R3,LOLIMW       ;CHECK IF LOWER THAN LOLIMIT
033310  103745                                  BLO     T3100$          ;YES - SKIP
033312  020337  013166                          CMP     R3,HILIMW       ;CHECK IF HIGHER THAN HILIMIT
033316  101342                                  BHI     T3100$          ;YES - SKIP
033320  012704  002322                          MOV     #T33TBL,R4      ;GET ADDRESS OF SEEK TABLE
033324  110364  000001                          MOVB    R3,1(R4)        ;INSERT CC IN APPROPRIATE TABLE
033330  110364  000003                          MOVB    R3,3(R4)        ; LOCATIONS FOR TEST SEEK SEQUENCE
033334  110364  000005                          MOVB    R3,5(R4)
033340  110364  000007                          MOVB    R3,7(R4)
033344  110364  000011                          MOVB    R3,11(R4)
033350  110364  000013                          MOVB    R3,13(R4)
033354  010437  002436                          MOV     R4,TBLSTR       ;STORE TABLE ADDRESS
033360  004737  017124                          JSR     PC,CHOSHD       ;GO CHOSE HEAD
033364                                  T3101$:
033364                                  BGNSUB
(3) 033364                                                                              T16.1:
```

```
  (3) 033364 104002                          EMT     C$BSUB
6254 033366 042737 003760 002416             BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
6255 033374 005737 002644                    TST     PASCNT          ;TEST IF PASS 0
6256 033400 001414                           BEQ     11$             ;YES - SKIP
6257 033402 023727 002644 000003             CMP     PASCNT,#3       ;TEST IF PASS 3
6258 033410 001404                           BEQ     10$             ;YES - SKIP
6259 033412 002407                           BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
6260 033414 012737 000003 002644             MOV     #3,PASCNT       ;ELSE SET TO 3
6261 033422 052737 000020 002416 10$:        BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
6262 033430 000405                           BR      12$             ;SKIP
6263 033432 005037 002644        11$:        CLR     PASCNT          ;SET PASS COUNT TO 0
6264 033436 052737 000040 002416             BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
6265 033444 012737 000003 002434 12$:        MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
6266 033452 013704 002436                    MOV     TBLSTR,R4       ;GET STORED TABLE ADDRESS
6267 033456 005037 002526        15$:        CLR     DESSEC          ;CLEAR TO SECTOR 0
6268 033462 112437 002514                    MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
6269 033466 004737 015624                    JSR     PC,XSEEK        ;DO SEEK
6270 033472 034036                           60$
6271 033474 012701 005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
6272 033500 004737 020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
6273 033504 034036                           60$
6274 033506 112437 002514                    MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
6275 033512 004737 015624                    JSR     PC,XSEEK        ;DO SEEK
6276 033516 034036                           60$
6277 033520 012701 005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
6278 033524 004737 020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
6279 033530 034036                           60$
6280 033532 004737 021000                    JSR     PC,VERPOS       ;VERIFY POSITION
6281 033536 034036                           60$
6282 033540 004737 022630        16$:        JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
6283 033544 033676                           32$                     ;"YES" RETURN
6284 033546 013737 002526 033566             MOV     DESSEC,25$      ;SET DATA PATTERN = TO SECTOR NUMBER
6285 033554 042737 177770 033566             BIC     #177770,25$     ;CLEAR ALL BUT LSD
6286 033562 004537 021462                    JSR     R5,DATGEN       ;GO GENERATE DATA
6287 033566 000000               25$:        .WORD   0
6288 033570 032737 000001 002434             BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
6289 033576 001425                           BEQ     29$             ;NO - SKIP
6290 033600 004737 022110                    JSR     PC,XWRITE       ;DO WRITE
6291 033604 034036                           60$
6292 033606 005237 002526                    INC     DESSEC          ;INC SECTOR
6293 033612 022737 000050 002526             CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
6294 033620 001347                           BNE     16$             ;NO - SKIP
6295 033622 042737 000060 002416             BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
6296 033630 042737 000001 002434             BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
6297 033636 052737 000100 002416             BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
6298 033644 005037 002526                    CLR     DESSEC          ;CLEAR TO SECTOR 0
6299 033650 000733                           BR      16$             ;SKIP
6300 033652 032737 000002 002434 29$:        BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
6301 033660 001414                           BEQ     33$             ;NO - SKIP
6302 033662 004737 022150        31$:        JSR     PC,XREAD        ;ELSE DO READ
6303 033666 034036                           60$
6304 033670 004737 021622                    JSR     PC,DATCOM       ;COMPARE DATA
6305 033674 034036                           60$
6306 033676 005237 002526        32$:        INC     DESSEC          ;BUMP SECTOR
6307 033702 022737 000050 002526             CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
6308 033710 001313                           BNE     16$             ;NO - LOOP
```

```
 6309  033712  005037  002526          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
 6310  033716  005037  002434                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
 6311  033722  005237  002644                  INC     PASCNT          ;BUMP PASS COUNT
 6312  033726  042737  003760  002416          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
 6313  033734  023727  002644  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
 6314  033742  001435                          BEQ     60$             ;YES - SKIP
 6315  033744  023727  002644  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
 6316  033752  001431                          BEQ     60$             ;YES - SKIP
 6317  033754  012737  000002  002434          MOV     #BIT1,WRTSWI    ;SET READ REQUIRED BIT
 6318  033762  023727  002644  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
 6319  033770  001415                          BEQ     40$             ;YES - SKIP
 6320  033772  023727  002644  000005          CMP     PASCNT,#5       ;TEST IF PASS 4
 6321  034000  001411                          BEQ     40$             ;YES - SKIP
 6322  034002  000404                          BR      39$             ;SKIP
 6323  034004  052737  002000  002416  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
 6324  034012  000407                          BR      36$             ;GO DO NEXT PASS
 6325  034014  052737  000020  002416  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
 6326  034022  000403                          BR      36$             ;SKIP
 6327  034024  052737  000040  002416  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
 6328  034032  000137  033456          36$:    JMP     15$             ;GO DO NEXT PASS
 6329  034036  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 6330  034044                          ENDSUB
  (3)  034044                          L10054:
  (3)  034044  104003                          EMT     C$ESUB
 6331  034046                                  ESCAPE  TST             ;EXIT TEST IF ERROR
  (3)  034046  104010                          EMT     C$ESCAPE
  (3)  034050  000054                          .WORD   L10053-.
 6332  034052  012737  000003  002434          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
 6333  034060  023727  002644  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
 6334  034066  001004                          BNE     45$             ;NO - SKIP
 6335  034070  012737  002330  002436          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
 6336  034076  000410                          BR      48$             ;GO START PASS 4
 6337  034100  005037  002644          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
 6338  034104  004737  017150                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 6339  034110  033224                          T3100$                  ;ABORT RETURN
 6340  034112  012737  002322  002436          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
 6341  034120  000137  033364          48$:    JMP     T3101$          ;GO DO HEAD 1
 6342  034124                          T3165$:
 6343  034124                          ENDTST
  (3)  034124                          L10053:
  (3)  034124  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 17         **WRITE LOCK ERROR AND DATA PROTECTION
034126                                  BGNTST                   ;TEST 17
(3)  034126                                                                                  T17::
034126  005737  002652                          TST     PASNUM           ;TEST IF FIRST PASS
034132  001003                                  BNE     2$               ;NO - SKIP
034134  005737  013162                          TST     MISWIW           ;TEST IF RUN MANUAL INTERVENTION
034140  100402                                  BMI     3$               ;YES - SKIP
034142  000137  035116                  2$:     JMP     T3265$           ;EXIT TST
034146                                  3$:
034146                                          BGNSUB
(3)  034146                                                                                  T17.1:
(3)  034146  104002                             EMT     C$BSUB
034150  012737  006405  002424                  MOV     #P2T17E,ERHEAD   ;SET ERROR HEADER
034156  004737  014714                          JSR     PC,TSTINT        ;INITIALIZE TEST
034162  004737  014732                          JSR     PC,GSTATR        ;CLEAR DRIVE
034166  034770                                  60$
034170  005037  002524                          CLR     DESHD            ;SET TO HEAD 0
034174  005037  002526                          CLR     DESSEC           ;SET TO SECTOR 0
034200  005037  002514                          CLR     NEWCYL           ;CLEAR TO CYLINDER 0
034204  004737  015624                          JSR     PC,XSEEK         ;DO SEEK
034210  034770                                  60$
034212  012701  005670                          MOV     #3000.,R1        ;SET WAIT FOR 300 MS
034216  004737  020404                          JSR     PC,RDYWAIT       ;WAIT FOR READY
034222  034770                                  60$
034224  004737  021000                          JSR     PC,VERPOS        ;VERIFY POSITIGN
034230  034770                                  60$
034232  032737  020000  002464                  BIT     #WLSTAT,T.MP     ;TEST IF WRITE LOCK SET
034240  001114                                  BNE     7$               ;YES - SKIP
034242  004537  021462                          JSR     R5,DATGEN        ;GENERATE DATA
034246  000007                                  #7                       ;PATTERN 7
034250  004737  022110                          JSR     PC,XWRITE        ;WRITE DATA
034254  034770                                  60$
034256  004737  022150                          JSR     PC,XREAD         ;READ DATA
034262  034770                                  60$
034264  004737  021622                          JSR     PC,DATCOM        ;CHECK DATA
034270  034770                                  60$
034272                                          PRINTF  #FMTOP1,#OPR004,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST SET WR
(13) 034272  005046                             CLR     -(SP)
(13) 034274  153716  002445                     BISB    RLDRV+1,(SP)
(12) 034300  012746  005423                     MOV     #DRVNAM,-(SP)
(11) 034304  013746  002440                     MOV     RLBAS,-(SP)
(10) 034310  012746  005412                     MOV     #BASADD,-(SP)
(9)  034314  012746  007040                     MOV     #OPR1A,-(SP)
(8)  034320  012746  007067                     MOV     #OPR004,-(SP)
(7)  034324  012746  010322                     MOV     #FMTOP1,-(SP)
(6)  034330  012746  000007                     MOV     #7,-(SP)
(3)  034334  010600                            MOV     SP,R0
(4)  034336  104017                             EMT     C$PNTF
(4)  034340  062706  000020                     ADD     #20,SP
034344  012701  000006                          MOV     #6.,R1           ;SET WAIT COUNT FOR 30 SECONDS
034350                                  5$.     WAITMS  #50.             ;CALL WAIT
(3)  034350  012700  000062                     MOV     #50.,R0
(3)  034354  104026                             EMT     C$WTM
034356  004737  014732                          JSR     PC,GSTATR        ;GET STATUS
034362  034770                                  60$
034364  032737  020000  002464                  BIT     #WLSTAT,T.MP     ;CHECK IF WRITE LOCK SET
```

```
 6384  034372  001037                        BNE     7$                      ;YES - SKIP
 6385  034374                                PRINTF  #FMT2,#BELL             ;RING BELL
  (8)  034374  012746  010167                MOV     #BELL,-(SP)
  (7)  034400  012746  010430                MOV     #FMT2,-(SP)
  (6)  034404  012746  000002                MOV     #2,-(SP)
  (3)  034410  010600                        MOV     SP,R0
  (4)  034412  104017                        EMT     C$PNTF
  (4)  034414  062706  000006                ADD     #6,SP
 6386  034420  005301                        DEC     R1                      ;DEC COUNT
 6387  034422  001352                        BNE     5$                      ;SKIP IF NOT 0
 6388  034424                                PRINTF  #FMT23,#P2T17E,#BYPSNM,#OPR1A,<B,RLDRV+1> ;RPT BYPASSED
  (11) 034424  005046                        CLR     -(SP)
  (11) 034426  153716  002445                BISB    RLDRV+1,(SP)
  (10) 034432  012746  007040                MOV     #OPR1A,-(SP)
  (9)  034436  012746  007143                MOV     #BYPSNM,-(SP)
  (8)  034442  012746  006405                MOV     #P2T17E,-(SP)
  (7)  034446  012746  011237                MOV     #FMT23,-(SP)
  (6)  034452  012746  000005                MOV     #5,-(SP)
  (3)  034456  010600                        MOV     SP,R0
  (4)  034460  104017                        EMT     C$PNTF
  (4)  034462  062706  000014                ADD     #14,SP
 6389  034466                                EXIT    TST
  (3)  034466  104032                        EMT     C$EXIT
  (3)  034470  000426                        .WORD   L10055-.
 6390  034472  004537  021462          7$:   JSR     R5,DATGEN               ;GENERATE DATA
 6391  034476  000001                        #1                              ;PATTERN 1
 6392  034500  012705  002446                MOV     #L.CS,R5                ;GET ADDRESS OF L REGS
 6393  034504  012715  000112                MOV     #WTDATA,(R5)            ;LOAD WRITE COMMAND
 6394  034510  053715  002444                BIS     RLDRV,(R5)              ;INSERT DRIVE NUMBER
 6395  034514  042725  002000                BIC     #BIT10,(R5)+            ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 6396  034520  012725  003656                MOV     #OBUFF,(R5)+            ;LOAD BUS ADDRESS
 6397  034524  005025                        CLR     (R5)+                   ;CYL 0, HD 0, SECTOR 0
 6398  034526  012725  177600                MOV     #177600,(R5)+           ;128 WORDS
 6399  034532  012701  000454                MOV     #300.,R1                ;SET WAIT COUNT FOR 30 MS
 6400  034536  005037  002420                CLR     DONE                    ;CLEAR INTERRUPT FLAG
 6401  034542  014562  000006                MOV     -(R5),RLMP(R2)          ;LOAD RL REGS
 6402  034546  014562  000004                MOV     -(R5),RLDA(R2)
 6403  034552  014562  000002                MOV     -(R5),RLBA(R2)
 6404  034556  014562  000000                MOV     -(R5),RLCS(R2)
 6405  034562                          10$:  WAITUS  #1
  (3)  034562  012700  000001                MOV     #1,R0
  (3)  034566  104027                        EMT     C$WTU
 6406  034570  005737  002420                TST     DONE                    ;CHECK IF INTERRUPT
 6407  034574  001012                        BNE     14$                     ;YES - SKIP
 6408  034576  005301                        DEC     R1                      ;DEC WAIT COUNT
 6409  034600  001370                        BNE     10$                     ;LOOP IF NOT 0
 6410  034602  004737  014562                JSR     PC,WAITIN               ;WAIT FOR INTERRUPT
 6411  034606  012603                        MOV     (SP)+,R3                ;GET RESULT MESSAGE
 6412  034610                                ERRHRD  1701.,,ERR1
  (3)  034610  104443                        TRAP    T$ERCODE
  (5)  034612  003245                        .WORD   1701
  (5)  034614  011344                        .WORD   ERR1
 6413  034616                                EXIT    SUB
  (3)  034616  104032                        EMT     C$EXIT
  (3)  034620  000156                        .WORD   L10056-.
 6414  034622  004737  014762          14$:  JSR     PC,GSTAT                ;GET STATUS
```

```
 6415  034626  034770                          60$
 6416  034630  032737  040000  002456          BIT     #DRVERR,T.CS    ;TEST IF ANY ERROR SET
 6417  034636  001005                          BNE     15$             ;YES - SKIP
 6418  034640  012703  007455                  MOV     #MDRERR,R3      ;SET RESULT MESSAGE POINTER
 6419  034644                                  ERRHRD  1702.,,ERR3     ;REPORT ERROR NOT SET
  (3)  034644  104443                          TRAP    T$ERCODE
  (5)  034646  003246                          .WORD   1702
  (5)  034650  011460                          .WORD   ERR3
 6420  034652  032737  002000  002464  15$:    BIT     #WGESTAT,T.MP   ;TEST IF WGE SET
 6421  034660  001005                          BNE     18$             ;YES - SKIP
 6422  034662  012703  007534                  MOV     #MWGERR,R3      ;SET MESSAGE FOR WGE NOT SET
 6423  034666                                  ERRHRD  1704.,,ERR3
  (3)  034666  104443                          TRAP    T$ERCODE
  (5)  034670  003250                          .WORD   1704
  (5)  034672  011460                          .WORD   ERR3
 6424  034674  042737  040000  002456  18$:    BIC     #DRVERR,T.CS    ;CLEAR DRIVE ERROR BIT
 6425  034702  042737  002000  002464          BIC     #WGESTAT,T.MP   ;CLEAR WGE BIT
 6426  034710  032737  157400  002464          BIT     #157400,T.MP    ;TEST IF ANY OTHER ERRORS
 6427  034716  001004                          BNE     16$             ;YES - GO REPORT
 6428  034720  032737  036000  002456          BIT     #36000,T.CS     ;TEST ANY ERRORS IN CS REG
 6429  034726  001404                          BEQ     17$             ;NO - SKIP
 6430  034730                          16$:    ERRHRD  1703.,,ERR6     ;REPORT ERRORS
  (3)  034730  104443                          TRAP    T$ERCODE
  (5)  034732  003247                          .WORD   1703
  (5)  034734  011646                          .WORD   ERR6
 6431  034736  000414                          BR      60$             ;EXIT TEST
 6432  034740  004737  014732          17$:    JSR     PC,GSTATR       ;GET STATUS AND RESET ERROR
 6433  034744  034770                          60$
 6434  034746  004537  021462                  JSR     R5,DATGEN       ;GO GENERATE DATA
 6435  034752  000007                          #7                      ;PATTERN 7
 6436  034754  004737  022150                  JSR     PC,XREAD        ;READ DATA
 6437  034760  034770                          60$
 6438  034762  004737  021622                  JSR     PC,DATCOM       ;COMPARE DATA
 6439  034766  034770                          60$
 6440  034770  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 6441  034776                          ENDSUB
  (3)  034776                          L10056:
  (3)  034776  104003                          EMT     C$ESUB
 6442  035000  012737  000002  002430  T3204$: MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 6443  035006                                  PRINTF  #FMTOP1,#OPR12,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQ RESET WRT L
  (13) 035006  005046                          CLR     -(SP)
  (13) 035010  153716  002445                  BISB    RLDRV+1,(SP)
  (12) 035014  012746  005423                  MOV     #DRVNAM,-(SP)
  (11) 035020  013746  002440                  MOV     RLBAS,-(SP)
  (10) 035024  012746  005412                  MOV     #BASADD,-(SP)
  (9)  035030  012746  007040                  MOV     #OPR1A,-(SP)
  (8)  035034  012746  007021                  MOV     #OPR12,-(SP)
  (7)  035040  012746  010322                  MOV     #FMTOP1,-(SP)
  (6)  035044  012746  000007                  MOV     #7,-(SP)
  (3)  035050  010600                          MOV     SP,R0
  (4)  035052  104017                          EMT     C$PNTF
  (4)  035054  062706  000020                  ADD     #20,SP
 6444  035060  012701  000454                  MOV     #300.,R1        ;SET WAIT FOR 30 SEC
 6445  035064                          16$:    WAITMS  #1
  (3)  035064  012700  000001                  MOV     #1,R0
  (3)  035070  104026                          EMT     C$WTM
```

```
 6446  035072  004737  014732                JSR     PC,GSTATR       ;GET STATUS
 6447  035076  035000                        T3204$
 6448  035100  032737  020000  002464        BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK RESET
 6449  035106  001403                        BEQ     T3265$
 6450  035110  005301                        DEC     R1              ;DEC WAIT COUNT
 6451  035112  001364                        BNE     16$             ;LOOP IF NOT 0
 6452  035114  000731                        BR      T3204$          ;ELSE REPEAT MESSAGE
 6453  035116                        T3265$:
 6454  035116                        ENDTST
  (3)  035116                        L10055:
  (3)  035116  104001                        EMT     C$ETST
 6455
```

```
                                        .SBTTL  *TEST 18         **ADJACENT CYLINDER INTERFERENCE
035120                                  BGNTST                   ;TEST 18
(3) 035120                                                                              T18::
035120  012737  006445  002424          MOV     #P2T18E,ERHEAD   ;SET ERROR HEADER
035126  004737  017234                  JSR     PC,CKBSVD        ;GO CHECK IF BAD SECTOR FILES VALID
035132  004737  014714                  JSR     PC,TSTINT        ;INITIALIZE TEST
035136  004737  014732                  JSR     PC,GSTATR        ;CLEAR DRIVE
035142  036226                          T3365$
035144  005037  002644                  CLR     PASCNT           ;CLEAR PASS TO 0
035150  012705  177777                  MOV     #-1,R5           ;SET R5
035154  005737  002652                  TST     PASNUM           ;TEST IF FIRST PASS (QUICK VERIFY)
035160  001007                          BNE     1$               ;NO - SKIP
035162  032737  000001  013162          BIT     #ALLCYL,MISWIW   ;TEST IF USE ALL CYLINDERS
035170  001003                          BNE     1$               ;YES - SKIP
035172  012705  177754                  MOV     #-20.,R5                 ;ELSE SET R5 TO NEG 20
035176  000402                          BR      9$               ;SKIP
035200  012705  177774          1$:     MOV     #-4,R5           ;ELSE SET FOR NEG 4
035204  012701  002322          9$:     MOV     #T33TBL,R1       ;GET ADDRESS OF WORK TABLE
035210  012703  000010                  MOV     #10,R3           ;SET CLEAR COUNT
035214  013721  013164          2$:     MOV     LOLIMW,(R1)+     ;CLEAR LOCATIONS TO LOLIMIT
035220  005303                          DEC     R3               ;DEC COUNT
035222  001374                          BNE     2$               ;LOOP UNTIL 0
035224  004537  021462                  JSR     R5,DATGEN        ;GO GENERATE DATA
035230  000011                          #9.                      ;PATTERN 9
035232  113737  013166  002324          MOVB    HILIMW,T33TBL+2  ;INSERT HILIMIT
035240  113737  013166  002326          MOVB    HILIMW,T33TBL+4  ;INTO APPROPRIATE LOCATIONS
035246  113737  013166  002332          MOVB    HILIMW,T33TBL+10
035254  113737  013166  002340          MOVB    HILIMW,T33TBL+16
035262  005205                  T3300$: INC     R5               ;BUMP R5
035264  032737  000001  013162          BIT     #ALLCYL,MISWIW   ;TEST IF USE ALL CYLINDERS
035272  001022                          BNE     5$               ;YES - SKIP
035274  005737  002652                  TST     PASNUM           ;TEST IF FIRST PASS (QUICK VERIFY)
035300  001403                          BEQ     3$               ;NO - SKIP
035302  062705  000003                  ADD     #3,R5            ;ELSE BUMP CYLINDER POINTER BY 3
035306  000402                          BR      6$               ;SKIP
035310  062705  000023          3$:     ADD     #19.,R5          ;BUMP TO NEXT ENTRY
035314  020527  000051          6$:     CMP     R5,#41.          ;CHECK IF PAST TABLE
035320  103005                          BHIS    4$               ;YES - SKIP TO EXIT
035322  116503  002342                  MOVB    CYLTBL(R5),R3    ;GET NEXT TABLE ENTRY
035326  042703  177400                  BIC     #177400,R3       ;CLEAR UPPER BYTE
035332  001011                          BNE     8$               ;SKIP IF NOT 0
035334  000137  036226          4$:     JMP     T3365$           ;EXIT TEST
035340  005705                  5$:     TST     R5               ;TEST IF R5 0
035342  001001                          BNE     7$               ;NO - SKIP
035344  005205                          INC     R5               ;ELSE BUMP R5 AGAIN
035346  022705  000377          7$:     CMP     #255.,R5         ;TEST IF ALL CYLINDERS USED
035352  001770                          BEQ     4$               ;YES - EXIT TEST
035354  010503                          MOV     R5,R3            ;USE R5 AS NEXT CYLINDER
035356  020337  013164          8$:     CMP     R3,LOLIMW        ;CHECK IF LOWER THAN LOLIMIT
035362  103737                          BLO     T3300$           ;YES - SKIP
035364  020337  013166                  CMP     R3,HILIMW        ;CHECK IF HIGHER THAN HILIMIT
035370  101334                          BHI     T3300$           ;YES - SKIP
035372  012704  002322                  MOV     #T33TBL,R4       ;GET ADDRESS OF SEEK TABLE
035376  110364  000001                  MOVB    R3,1(R4)         ;INSERT CC IN APPROPRIATE TABLE
035402  110364  000007                  MOVB    R3,7(R4)
035406  110364  000011                  MOVB    R3,11(R4)
```

```
 6512  035412  110364  000017                    MOVB    R3,17(R4)
 6513  035416  005203                            INC     R3              ;BUMP R3 TO CC+1
 6514  035420  110364  000005                    MOVB    R3,5(R4)        ;INSERT AS NEEDED
 6515  035424  110364  000015                    MOVB    R3,15(R4)
 6516  035430  162703  000002                    SUB     #2,R3           ;SET R3 TO CC-1
 6517  035434  110364  000003                    MOVB    R3,3(R4)        ;INSERT AS NEEDED
 6518  035440  110364  000013                    MOVB    R3,13(R4)
 6519  035444  010437  002436                    MOV     R4,TBLSTR       ;STORE TABLE ADDRESS
 6520  035450  004737  017124                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
 6521  035454                          T3301$:
 6522  035454                          BGNSUB
  (3)  035454                                                            T18.1:
  (3)  035454  104002                            EMT     C$BSUB
 6523  035456  042737  003760  002416            BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
 6524  035464  005737  002644                    TST     PASCNT          ;TEST IF PASS 0
 6525  035470  001414                            BEQ     11$             ;YES - SKIP
 6526  035472  023727  002644  000004            CMP     PASCNT,#4       ;TEST IF PASS 4
 6527  035500  001404                            BEQ     10$             ;YES - SKIP
 6528  035502  002407                            BLT     11$             ;CHECK IF LESS THAN 4, IF YES CLEAR TO 0
 6529  035504  012737  000004  002644            MOV     #4,PASCNT       ;ELSE SET TO 4
 6530  035512  052737  000020  002416  10$:       BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
 6531  035520  000405                            BR      12$             ;SKIP
 6532  035522  005037  002644            11$:    CLR     PASCNT          ;SET PASS COUNT TO 0
 6533  035526  052737  000040  002416            BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
 6534  035534  012737  000003  002434  12$:      MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
 6535  035542  013704  002436                    MOV     TBLSTR,R4       ;GET STORED TABLE ADDRESS
 6536  035546  005037  002526            15$:    CLR     DESSEC          ;CLEAR TO SECTOR 0
 6537  035552  112437  002514                    MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
 6538  035556  004737  015624                    JSR     PC,XSEEK        ;DO SEEK
 6539  035562  036140                            60$
 6540  035564  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 6541  035570  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
 6542  035574  036140                            60$
 6543  035576  112437  002514                    MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
 6544  035602  004737  015624                    JSR     PC,XSEEK        ;DO SEEK
 6545  035606  036140                            60$
 6546  035610  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 6547  035614  004737  020404                    JSR     PC,RDYWAIT      ;WAIT FOR READY
 6548  035620  036140                            60$
 6549  035622  004737  021000                    JSR     PC,VERPOS       ;VERIFY POSITION
 6550  035626  036140                            60$
 6551  035630  004737  022630          16$:      JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
 6552  035634  035744                            32$                     ;"YES" RETURN
 6553  035636  032737  000001  002434            BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
 6554  035644  001425                            BEQ     29$             ;NO - SKIP
 6555  035646  004737  022110                    JSR     PC,XWRITE       ;DO WRITE
 6556  035652  036140                            60$
 6557  035654  005237  002526                    INC     DESSEC          ;INC SECTOR
 6558  035660  022737  000050  002526            CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 6559  035666  001360                            BNE     16$             ;NO - SKIP
 6560  035670  042737  000060  002416            BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
 6561  035676  042737  000001  002434            BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
 6562  035704  052737  000100  002416            BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
 6563  035712  005037  002526                    CLR     DESSEC          ;CLEAR TO SECTOR 0
 6564  035716  000744                            BR      16$             ;SKIP
 6565  035720  032737  000002  002434  29$:      BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
```

```
 6566  035726  001414                          BEQ     33$             ;NO - SKIP
 6567  035730  004737  022150          31$:    JSR     PC,XREAD        ;ELSE DO READ
 6568  035734  036140                          60$
 6569  035736  004737  021622                  JSR     PC,DATCOM       ;COMPARE DATA
 6570  035742  036140                          60$
 6571  035744  005237  002526          32$:    INC     DESSEC          ;BUMP SECTOR
 6572  035750  022737  000050  002526          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 6573  035756  001324                          BNE     16$             ;NO - LOOP
 6574  035760  005037  002526          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
 6575  035764  005037  002434                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
 6576  035770  005237  002644                  INC     PASCNT          ;BUMP PASS COUNT
 6577  035774  042737  003760  002416          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
 6578  036002  023727  002644  000004          CMP     PASCNT,#4       ;TEST IS PASS 4
 6579  036010  001453                          BEQ     60$             ;YES - SKIP
 6580  036012  023727  002644  000010          CMP     PASCNT,#8.      ;TEST IF PASS 8.
 6581  036020  001447                          BEQ     60$             ;YES - SKIP
 6582  036022  023727  002644  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
 6583  036030  001430                          BEQ     39$             ;YES - SKIP
 6584  036032  023727  002644  000007          CMP     PASCNT,#7       ;TEST IF PASS 7
 6585  036040  001430                          BEQ     40$             ;YES - SKIP
 6586  036042  012737  000001  002434          MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED
 6587  036050  023727  002644  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
 6588  036056  001411                          BEQ     37$             ;YES - SKIP
 6589  036060  023727  002644  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
 6590  036066  001405                          BEQ     37$             ;YES - SKIP
 6591  036070  052737  000040  002416          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
 6592  036076  000137  035546          36$:    JMP     15$             ;GO DO NEXT PASS
 6593  036102  052737  000020  002416  37$:    BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUALIFIER
 6594  036110  000772                          BR      36$
 6595  036112  052737  000200  002416  39$:    BIS     #REVSKS,OPFLAG  ;SET MESSAGE QUALIFIER
 6596  036120  000403                          BR      41$
 6597  036122  052737  000400  002416  40$:    BIS     #FWDSKS,OPFLAG  ;SET MESSAGE QUALIFIER
 6598  036130  012737  000002  002434  41$:    MOV     #BIT1,WRTSWI    ;SET READ REQUIRED
 6599  036136  000757                          BR      36$
 6600  036140  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 6601  036146                          ENDSUB
  (3)  036146                          L10060:
  (3)  036146  104003                          EMT     C$ESUB
 6602  036150                          ESCAPE  TST                     ;EXIT TEST IF ERROR
  (3)  036150  104010                          EMT     C$ESCAPE
  (3)  036152  000054                          .WORD   L10057-.
 6603  036154  012737  000003  002434          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
 6604  036162  023727  002644  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
 6605  036170  001004                          BNE     45$             ;NO - SKIP
 6606  036172  012737  002332  002436          MOV     #T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
 6607  036200  000410                          BR      48$             ;GO START PASS 4
 6608  036202  005037  002644          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
 6609  036206  004737  017150                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 6610  036212  035262                          T3300$                  ;ABORT RETURN
 6611  036214  012737  002322  002436          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
 6612  036222  000137  035454          48$:    JMP     T3301$          ;GO DO HEAD 1
 6613  036226                          T3365$:
 6614  036226                          ENDTST
  (3)  036226                          L10057:
  (3)  036226  104001                          EMT     C$ETST
```

```
6616                                    .SBTTL  *TEST 19         **OVERWRITE
6617  036230                            BGNTST                   ;TEST 19
 (3)  036230                                                                 T19::
6618  036230  012737  006472  002424            MOV     #P2T19E,ERHEAD  ;SET ERROR HEADER
6619  036236  004737  017234                    JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
6620  036242  004737  014714                    JSR     PC,TSTINT       ;INITIALIZE TEST
6621  036246  004737  014732                    JSR     PC,GSTATR       ;CLEAR DRIVE
6622  036252  037334                            T3465$
6623  036254  005037  002644                    CLR     PASCNT          ;CLEAR PASS TO 0
6624  036260  012705  177777                    MOV     #-1,R5          ;SET R5
6625  036264  005737  002652                    TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
6626  036270  001007                            BNE     1$              ;NO - SKIP
6627  036272  032737  000001  013162            BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
6628  036300  001003                            BNE     1$              ;YES - SKIP
6629  036302  012705  177754                    MOV     #-20.,R5        ;ELSE SET R5 TO NEG 20
6630  036306  000402                            BR      9$              ;SKIP
6631  036310  012705  177774          1$:       MOV     #-4,R5          ;SET FOR NEXT ENTRY
6632  036314  012701  002322          9$:       MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
6633  036320  012703  000010                    MOV     #10,R3          ;SET CLEAR COUNT
6634  036324  013721  013164          2$:       MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
6635  036330  005303                            DEC     R3              ;DEC COUNT
6636  036332  001374                            BNE     2$              ;LOOP UNTIL 0
6637  036334  113737  013166  002324            MOVB    HILIMW,T33TBL+2 ;INSERT HILIMIT
6638  036342  113737  013166  002330            MOVB    HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
6639  036350  113737  013166  002334            MOVB    HILIMW,T33TBL+12
6640  036356  005205                  T3400$:   INC     R5              ;BUMP R5
6641  036360  032737  000001  013162            BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
6642  036366  001022                            BNE     5$              ;YES - SKIP
6643  036370  005737  002652                    TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
6644  036374  001003                            BNE     3$              ;NO - SKIP
6645  036376  062705  000023                    ADD     #19.,R5         ;ELSE BUMP CYLINDER POINTER BY 19
6646  036402  000402                            BR      6$              ;SKIP
6647  036404  062705  000003          3$:       ADD     #3,R5           ;BUMP CYLINDER POINTER BY 3
6648  036410  020527  000051          6$:       CMP     R5,#41.         ;TEST IF PAST VALID TABLE
6649  036414  103005                            BHIS    4$              ;YES - SKIP
6650  036416  116503  002342                    MOVB    CYLTBL(R5),R3   ;GET NEXT TABLE ENTRY
6651  036422  042703  177400                    BIC     #177400,R3      ;CLEAR UPPER BYTE
6652  036426  001011                            BNE     8$              ;SKIP IF NOT 0
6653  036430  000137  037334          4$:       JMP     T3465$          ;EXIT TEST
6654  036434  005705                5$:       TST     R5              ;TEST IF R5 0
6655  036436  001001                            BNE     7$              ;NO - SKIP
6656  036440  005205                            INC     R5              ;ELSE BUMP R5 AGAIN
6657  036442  022705  000377          7$:       CMP     #255.,R5        ;TEST IF ALL CYLINDERS USED
6658  036446  001770                            BEQ     4$              ;YES - EXIT TEST
6659  036450  010503                            MOV     R5,R3           ;USE R5 AS NEXT CYLINDER
6660  036452  020337  013164          8$:       CMP     R3,LOLIMW       ;TEST IF PAST LO LIMIT
6661  036456  103737                            BLO     T3400$          ;YES - SKIP
6662  036460  020337  013166                    CMP     R3,HILIMW       ;TEST IF PAST HILIMIT
6663  036464  101334                            BHI     T3400$          ;YES - SKIP
6664  036466  012704  002322                    MOV     #T33TBL,R4      ;GET ADDRESS OF SEEK TABLE
6665  036472  110364  000001                    MOVB    R3,1(R4)        ;INSERT CC IN APPROPRIATE TABLE
6666  036476  110364  000003                    MOVB    R3,3(R4)        ; LOCATIONS FOR TEST SEEK SEQUENCE
6667  036502  110364  000005                    MOVB    R3,5(R4)
6668  036506  110364  000007                    MOVB    R3,7(R4)
6669  036512  110364  000011                    MOVB    R3,11(R4)
6670  036516  110364  000013                    MOVB    R3,13(R4)
```

```
 6671  036522  010437  002436                MOV     R4,TBLSTR       ;STORE TABLE ADDRESS
 6672  036526  004737  017124                JSR     PC,CHOSHD       ;GO CHOSE HEAD
 6673  036532                        T3401$:
 6674  036532                        BGNSUB
  (3)  036532                                                        T19.1:
  (3)  036532  104002                        EMT     C$BSUB
 6675  036534  042737  003760  002416        BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
 6676  036542  005737  002644                TST     PASCNT          ;TEST IF PASS 0
 6677  036546  001414                        BEQ     11$             ;YES - SKIP
 6678  036550  023727  002644  000003        CMP     PASCNT,#3       ;TEST IF PASS 3
 6679  036556  001404                        BEQ     10$             ;YES - SKIP
 6680  036560  002407                        BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
 6681  036562  012737  000003  002644        MOV     #3,PASCNT       ;ELSE SET TO 3
 6682  036570  052737  000020  002416  10$:   BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
 6683  036576  000405                        BR      12$             ;SKIP
 6684  036600  005037  002644          11$:   CLR     PASCNT          ;SET PASS COUNT TO 0
 6685  036604  052737  000040  002416        BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
 6686  036612  012737  000003  002434  12$:   MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
 6687  036620  013704  002436                MOV     TBLSTR,R4       ;GET STORED TABLE ADDRESS
 6688  036624  005037  002526          15$:   CLR     DESSEC          ;CLEAR TO SECTOR 0
 6689  036630  112437  002514                MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
 6690  036634  004737  015624                JSR     PC,XSEEK        ;DO SEEK
 6691  036640  037246                        60$
 6692  036642  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 6693  036646  004737  020404                JSR     PC,RDYWAIT      ;WAIT FOR READY
 6694  036652  037246                        60$
 6695  036654  112437  002514                MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
 6696  036660  004737  015624                JSR     PC,XSEEK        ;DO SEEK
 6697  036664  037246                        60$
 6698  036666  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 6699  036672  004737  020404                JSR     PC,RDYWAIT      ;WAIT FOR READY
 6700  036676  037246                        60$
 6701  036700  004737  021000                JSR     PC,VERPOS       ;VERIFY POSITION
 6702  036704  037246                        60$
 6703  036706  004737  022630          16$:   JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
 6704  036712  037062                        32$                     ;"YES" RETURN
 6705  036714  005737  002644                TST     PASCNT          ;TEST IF PASS 0
 6706  036720  001407                        BEQ     17$             ;YES - SKIP
 6707  036722  022737  000003  002644        CMP     #3,PASCNT       ;TEST IF PASS 3
 6708  036730  001403                        BEQ     17$             ;YES - SKIP
 6709  036732  005037  036752                CLR     25$             ;ELSE CLEAR DATA PATTERN SELECTOR
 6710  036736  000403                        BR      18$
 6711  036740  012737  000010  036752  17$:   MOV     #8.,25$         ;SET DATA PATTERN SELECTOR TO 8
 6712  036746  004537  021462          18$:   JSR     R5,DATGEN       ;GO GENERATE DATA
 6713  036752  000000          25$:   .WORD   0
 6714  036754  032737  000001  002434        BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
 6715  036762  001425                        BEQ     23$             ;NO - SKIP
 6716  036764  004737  022110                JSR     PC,XWRITE       ;DO WRITE
 6717  036770  037246                        60$
 6718  036772  005237  002526                INC     DESSEC          ;INC SECTOR
 6719  036776  022737  000050  002526        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 6720  037004  001340                        BNE     16$             ;NO - SKIP
 6721  037006  042737  000060  002416        BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
 6722  037014  042737  000001  002434        BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
 6723  037022  052737  000100  002416        BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
 6724  037030  005037  002526                CLR     DESSEC          ;CLEAR TO SECTOR 0
```

```
 6725  037034  000724                          BR      16$             ;SKIP
 6726  037036  032737  000002  002434  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
 6727  037044  001414                          BEQ     33$             ;NO - SKIP
 6728  037046  004737  022150          31$:    JSR     PC,XREAD        ;ELSE DO READ
 6729  037052  037246                          60$
 6730  037054  004737  021622                  JSR     PC,DATCOM       ;COMPARE DATA
 6731  037060  037246                          60$
 6732  037062  005237  002526          32$:    INC     DESSEC          ;BUMP SECTOR
 6733  037066  022737  000050  002526          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 6734  037074  001304                          BNE     16$             ;NO - LOOP
 6735  037076  005037  002526          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
 6736  037102  005037  002434                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
 6737  037106  005237  002644                  INC     PASCNT          ;BUMP PASS COUNT
 6738  037112  042737  003760  002416          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
 6739  037120  023727  002644  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
 6740  037126  001447                          BEQ     60$             ;YES - SKIP
 6741  037130  023727  002644  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
 6742  037136  001443                          BEQ     60$             ;YES - SKIP
 6743  037140  023727  002644  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
 6744  037146  001424                          BEQ     39$             ;YES - SKIP
 6745  037150  023727  002644  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
 6746  037156  001424                          BEQ     40$             ;YES - SKIP
 6747  037160  012737  000002  002434          MOV     #BIT1,WRTSWI    ;SET WRITE REQUIRED BIT
 6748  037166  023727  002644  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
 6749  037174  001405                          BEQ     37$             ;YES - SKIP
 6750  037176  052737  001000  002416          BIS     #REVSKO,OPFLAG  ;SET REVERSE QUALIFIER
 6751  037204  000137  036624          36$:    JMP     15$             ;GO DO NEXT PASS
 6752  037210  052737  002000  002416  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
 6753  037216  000772                          BR      36$             ;GO DO NEXT PASS
 6754  037220  052737  000020  002416  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
 6755  037226  000403                          BR      41$             ;SKIP
 6756  037230  052737  000040  002416  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
 6757  037236  012737  000001  002434  41$:    MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED BIT
 6758  037244  000757                          BR      36$             ;GO DO NEXT PASS
 6759  037246  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 6760  037254                          ENDSUB
  (3)  037254                          L10062:
  (3)  037254  104003                          EMT     C$ESUB
 6761  037256                                  ESCAPE  TST             ;EXIT TEST IF ERROR
  (3)  037256  104010                          EMT     C$ESCAPE
  (3)  037260  000054                          .WORD   L10061-.
 6762  037262  012737  000003  002434          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
 6763  037270  023727  002644  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
 6764  037276  001004                          BNE     45$             ;NO - SKIP
 6765  037300  012737  002330  002436          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
 6766  037306  000410                          BR      48$             ;GO START PASS 4
 6767  037310  005037  002644          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
 6768  037314  004737  017150                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD ONE OR ABORT TEST
 6769  037320  036356                          T3400$                  ;ABORT RETURN
 6770  037322  012737  002322  002436          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
 6771  037330  000137  036532          48$:    JMP     T3401$          ;GO DO HEAD 1
 6772  037334                          T3465$:
 6773  037334                          ENDTST
  (3)  037334                          L10061:
  (3)  037334  104001                          EMT     C$ETST
 6774  037336                          ENDMOD
```

```
 6776  037336                          BGNMOD  HRDPRM
 6777  037336                          BGNHRD
  (3)  037336  000025                          .WORD L10063-L$HARD/2
 6778  037340                          GPRML   CNTYPE,CNT,1,YES
  (4)  037340  004130                          .WORD   T$CODE
  (4)  037342  037454                          .WORD   CNTYPE
  (4)  037344  000001                          .WORD   1
 6779  037346                          GPRMA   CSRMSG,CSR,0,160000,177776,YES
  (4)  037346  000031                          .WORD   T$CODE
  (4)  037350  037412                          .WORD   CSRMSG
  (4)  037352  160000                          .WORD   T$LOLIM
  (4)  037354  177776                          .WORD   T$HILIM
 6780  037356                          GPRMA   VECMSG,VECT,0,0,776,YES
  (4)  037356  001031                          .WORD   T$CODE
  (4)  037360  037426                          .WORD   VECMSG
  (4)  037362  000000                          .WORD   T$LOLIM
  (4)  037364  000776                          .WORD   T$HILIM
 6781  037366                          GPRMD   BRMSG,PRIOR,0,340,0,7,YES
  (4)  037366  002032                          .WORD   T$CODE
  (4)  037370  037435                          .WORD   BRMSG
  (4)  037372  000340                          .WORD   340
  (4)  037374  000000                          .WORD   T$LOLIM
  (4)  037376  000007                          .WORD   T$HILIM
 6782  037400                          GPRMD   DRMSG,DRSB,0,3400,0,7,YES
  (4)  037400  003032                          .WORD   T$CODE
  (4)  037402  037446                          .WORD   DRMSG
  (4)  037404  003400                          .WORD   3400
  (4)  037406  000000                          .WORD   T$LOLIM
  (4)  037410  000007                          .WORD   T$HILIM
 6783
 6784  037412                          ENDHRD
  (2)                                         .EVEN
  (3)  037412                          L10063:
 6785
 6786  037412  052502  020123  042101  CSRMSG: .ASCIZ  /BUS ADDRESS/
       037420  051104  051505  000123
 6787  037426  042526  052103  051117  VECMSG: .ASCIZ  /VECTOR/
       037434     000
 6788  037435     102  020122  042514  BRMSG:  .ASCIZ  /BR LEVEL/
       037442  042526  000114
 6789  037446  051104  053111  000105  DRMSG:  .ASCIZ  /DRIVE/
 6790  037454  046122  030461     000  CNTYPE: .ASCIZ  /RL11/
 6791  037461                          ENDMOD
 6792          037462                          .EVEN
 6793
 6794  037462                          BGNMOD  SFTPRM
 6795  037462                          BGNSFT
  (3)  037462  000061                          .WORD L10064-L$SOFT/2
 6796
 6798  037464                          GPRML   CYLQ,MISWI,1,YES
  (4)  037464  000130                          .WORD   T$CODE
  (4)  037466  037626                          .WORD   CYLQ
  (4)  037470  000001                          .WORD   1
 6799  037472                          GPRML   SECQ,MISWI,2,YES
  (4)  037472  000130                          .WORD   T$CODE
  (4)  037474  037650                          .WORD   SECQ
```

```
                                                .WORD   2
6805    037500                          GPRML   MANQ,MISWI,100000,YES
 (4)    037500  000130                         .WORD   T$CODE
 (4)    037502  037670                         .WORD   MANQ
 (4)    037504  100000                         .WORD   100000
6806
6808    037506                          GPRML   LOLIMQ,MISWI,40000,YES
 (4)    037506  000130                         .WORD   T$CODE
 (4)    037510  037732                         .WORD   LOLIMQ
 (4)    037512  040000                         .WORD   40000
6809    037514                                  XFERF   1$
 (5)    037514  006044                         .WORD   T$CODE
6810    037516                                  GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
 (4)    037516  001052                         .WORD   T$CODE
 (4)    037520  037753                         .WORD   LIMVAL
 (4)    037522  000377                         .WORD   255.
 (4)    037524  000000                         .WORD   T$LOLIM
 (4)    037526  000375                         .WORD   T$HILIM
6811    037530                          1$:     GPRML   HILIMQ,MISWI,20000,YES
 (4)    037530  000130                         .WORD   T$CODE
 (4)    037532  037767                         .WORD   HILIMQ
 (4)    037534  020000                         .WORD   20000
6812    037536                                  XFERF   2$
 (5)    037536  006044                         .WORD   T$CODE
6813    037540                                  GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
 (4)    037540  002052                         .WORD   T$CODE
 (4)    037542  037753                         .WORD   LIMVAL
 (4)    037544  000377                         .WORD   255.
 (4)    037546  000000                         .WORD   T$LOLIM
 (4)    037550  000377                         .WORD   T$HILIM
6814    037552                          2$:     GPRML   HEADQ,MISWI,10000,YES
 (4)    037552  000130                         .WORD   T$CODE
 (4)    037554  040010                         .WORD   HEADQ
 (4)    037556  010000                         .WORD   10000
6815    037560                                  XFERF   3$
 (5)    037560  006044                         .WORD   T$CODE
6816    037562                                  GPRMD   HEADV,HEAD,D,17,0,1,YES
 (4)    037562  003052                         .WORD   T$CODE
 (4)    037564  040035                         .WORD   HEADV
 (4)    037566  000017                         .WORD   17
 (4)    037570  000000                         .WORD   T$LOLIM
 (4)    037572  000001                         .WORD   T$HILIM
6818    037574                          3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
 (4)    037574  004052                         .WORD   T$CODE
 (4)    037576  040066                         .WORD   ERLIMQ
 (4)    037600  000377                         .WORD   377
 (4)    037602  000000                         .WORD   T$LOLIM
 (4)    037604  000377                         .WORD   T$HILIM
6820    037606                                  GPRMD   DCLIMQ,DCLIM,D,377,1,377,YES
 (4)    037606  005052                         .WORD   T$CODE
 (4)    037610  040144                         .WORD   DCLIMQ
 (4)    037612  000377                         .WORD   377
 (4)    037614  000001                         .WORD   T$LOLIM
 (4)    037616  000377                         .WORD   T$HILIM
6822    037620                          GPRML   AUTOQ,MISWI,20,YES
 (4)    037620  000130                         .WORD   T$CODE
```

```
  (4)  037622  040112                          .WORD   AUTOQ
  (4)  037624  000020                          .WORD   20
 6823  037626                        ENDSFT
  (2)                                         .EVEN
  (3)  037626                        L10064:
 6824
 6826  037626  051525  020105  046101  CYLQ:   .ASCIZ  /USE ALL CYLINDERS/
       037634  020114  054503  044514
       037642  042116  051105  000123
 6827  037650  051525  020105  046101  SECQ:   .ASCIZ  /USE ALL SECTORS/
       037656  020114  042523  052103
       037664  051117  000123
 6833  037670  054105  041505  052125  MANQ:   .ASCIZ  /EXECUTE MANUAL INTERVENTION TESTS/
       037676  020105  040515  052516
       037704  046101  044440  052116
       037712  051105  042526  052116
       037720  047511  020116  042524
       037726  052123  000123
 6835  037732  047514  042527  020122  LOLIMQ: .ASCIZ  /LOWER SEEK LIMIT/
       037740  042523  045505  046040
       037746  046511  052111     000
 6836  037753     105  052116  051105  LIMVAL: .ASCIZ  /ENTER VALUE/
       037760  053040  046101  042525
       037766     000
 6837  037767     125  050120  051105  HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
       037774  051440  042505  020113
       040002  044514  044515  000124
 6838  040010  051525  020105  047117  HEADQ:  .ASCIZ  /USE ONLY ONE SURFACE/
       040016  054514  047440  042516
       040024  051440  051125  040506
       040032  042503     000
 6839  040035     123  042520  044503  HEADV:  .ASCIZ  /SPECIFY SURFACE (0 OR 1)/
       040042  054506  051440  051125
       040050  040506  042503  024040
       040056  020060  051117  030440
       040064  000051
 6841  040066  050123  041505  043111  ERLIMQ: .ASCIZ  /SPECIFY ERROR LIMIT/
       040074  020131  051105  047522
       040102  020122  044514  044515
       040110  000124
 6842  040112  051104  050117  042040  AUTOQ:  .ASCIZ  /DROP DRIVE IF NO RESPONSE/
       040120  044522  042526  044440
       040126  020106  047516  051040
       040134  051505  047520  051516
       040142  000105
 6844  040144  040504  040524  041440  DCLIMQ: .ASCIZ  /DATA COMPARE ERROR LIMIT/
       040152  046517  040520  042522
       040160  042440  051122  051117
       040166  046040  046511  052111
       040174     000
 6846          040176                         .EVEN
 6847  040176                        ENDMOD
 6848  040176                        LASTAD
  (2)                                         .EVEN
```

```
 6849                                    .TITLE  PDP-11 DIAGNOSTIC SUPERVISOR
18145           071076                  END.SUPV=.+2
18147           071776                  .=71776
18148  071776   000000                  .WORD   0
18150           072000                  X1X1=.
18151           000200                  .END 200
```

```
ABOFLA  040450 G        BLOCK   063250          CRDYMS= 000200          C$PNTS= 000016          DSPCOD  013176 G
ABOPAS  040362 G        BRMSG   037435          CRLF    057626          C$PNTX= 000015          DUNIT.  040366 G
ABO.FM  043324          BSCHK   022630          CSNAM   005523          C$POIN= 000040          DVC.FT  053532
AFMID   002622          BSFLAG  002432          CSR   = 000000          C$Q10 = 000377          D$AAG   054402
AFMIDU  002624          BSFVAL  002664          CSRMSG  037412          C$RDBU= 000007          D$AAH   054420
ALLCYL= 000001          BSNSTR  007232          CURCYL  002516          C$REFG= 000050          D$AAI   057172
ALLOC   061114          BYPSNM  007143          CURR.T  040376 G        C$REQT= 000045          D$AAJ   057176
ALLSEC= 000002          B$AAB   047516          CYLQ    037626          C$RESE= 000033          D$AAK   057214
ANYERR= 100000          B$AAF   047430          CYLTBL  002342          C$REVI= 000001          D$AAL   057232
ARMID   002626          CAFDT   010300          CYLUP = 000004          C$RPT = 000025          D$AAM   057242
ARMIDU  002630          CALLPC= 000022          CYLWD   007130          C$SEFG= 000047          EF.CON= 000036 G
AUTOQ   040112          CALLPS= 000024          C$AAD   052732          C$SPRI= 000041          EF.NEW= 000035 G
AUTOSZ= 000020          CALLSP= 000026          C$AAE   052744          C$SVEC= 000037          EF.PWR= 000034 G
A$AAW   045234          CALLTC= 000030          C$AAK   053562          C$TPRI= 000013          EF.RES= 000037 G
A$AAX   045250          CAL.CL  065632          C$AAL   053672          C$UNBU= 000031          EF.STA= 000040 G
A$AAY   045256          CAL.TI  065670 G        C$ABRT= 000021          C$WTM = 000026          EF01  = 000001 G
A$AAZ   045272          CAMSK = 077600          C$ADR = 000020          C$WTU = 000027          EF02  = 000002 G
A$ABA   045302          CCYLUP  010267          C$AU  = 000054          C10MS   010246          EF03  = 000003 G
BADADD= 004000          CHKLUP  047532          C$BRK = 000022          C5SEC   010312          EF04  = 000004 G
BAMSK = 000060          CHKSTR  061456          C$BSEG= 000004          C500MS  010261          EF05  = 000005 G
BANAM   005530          CHKTTY  057544          C$BSUB= 000002          DANAM   005535          EF06  = 000006 G
BASADD  005412          CHK.FO  041742          C$BUFF= 000030          DATACM= 000001          EF07  = 000007 G
BELL    010167          CHK.MA  045672          C$CEFG= 000046          DATCOM  021622          EF08  = 000010 G
BGN.SU= 040176          CHK.PC  052760          C$CLEA= 000012          DATGEN  021462          EF09  = 000011 G
BHSTAT= 000010          CHK.SW  041464          C$CLP1= 000006          DCKERR= 004000          EF10  = 000012 G
BINMSG  057424          CHOSHD  017124          C$CVEC= 000036          DCLIM = 000012          EF11  = 000013 G
BIT0  = 000001 G        CHRCNT  060776          C$DCLN= 000044          DCLIMQ  040144          EF12  = 000014 G
BIT00 = 000001 G        CH.FLA  045374          C$DODU= 000053          DCLIMW  013174          EF13  = 000015 G
BIT01 = 000002 G        CH.PAS  045416          C$DRPT= 000024          DECMSG  057440          EF14  = 000016 G
BIT02 = 000004 G        CKBSVD  017234          C$DU  = 000055          DESDIF  002520          EF15  = 000017 G
BIT03 = 000010 G        CKDATA= 000102          C$EDIT= 000006          DESHD   002524          EF16  = 000020 G
BIT04 = 000020 G        CKERLM  014370          C$ERDF= 000002          DESSEC  002526          EMT.TR  040454 G
BIT05 = 000040 G        CLEAR.  047014          C$ERHR= 000003          DESSGN  002522          END.OF  047002
BIT06 = 000100 G        CLKACC  040360 G        C$ERSF= 000001          DIAG.T  040456 G        END.SU= 071076
BIT07 = 000200 G        CLKBFR  065634          C$ERSO= 000004          DIFAUG  002510          EOP.CH  067554 G
BIT08 = 000400 G        CLKCNT  040356 G        C$ESCA= 000010          DIFWD   007104          EOP.FM  043340
BIT09 = 001000 G        CLKRES  067232 G        C$ESEG= 000005          DIRBIT= 000004          EOP.IN  045412
BIT1  = 000002 G        CLKSER  067532 G        C$ESUB= 000003          DIRMSK= 077600          ERHEAD  002424
BIT10 = 002000 G        CLKSON  040422 G        C$ETST= 000001          DLTERR= 010000          ERLIM = 000010
BIT11 = 004000 G        CLK.SE  045474          C$EXIT= 000032          DONE    002420          ERLIMQ  040066
BIT12 = 010000 G        CLNCOD  014200 G        C$GMAN= 000043          DPDVD   070242 G        ERLIMW  013172
BIT13 = 020000 G        CLRPAR  024230          C$GPHR= 000042          DPMUL   070130 G        ERRCNT  002650
BIT14 = 040000 G        CLR.MA  045750          C$GPRI= 000040          DRDYMS= 000001          ERRFOR  053750
BIT15 = 100000 G        CNT   = 000010          C$GTIM= 000052          DRMSG   037446          ERRHAN  052764
BIT2  = 000004 G        CNTYPE  037454          C$INIT= 000011          DRSB  = 000006          ERRSWI  002430
BIT3  = 000010 G        CNVT    063710          C$INLP= 000020          DRSELT= 000004          ERRVEC  002642
BIT4  = 000020 G        COMMTA  063530          C$KWOF= 000035          DRSET = 000010          ERR.HR  053542
BIT5  = 000040 G        COMPOP= 007777          C$KWON= 000034          DRVCNT  002506          ERR.SF  053546
BIT6  = 000100 G        CONHNG= 000004          C$LOOP= 000100          DRVERR= 040000          ERR1    011344 G
BIT7  = 000200 G        CONTCL  067312 G        C$MANI= 000051          DRVNAM  005423          ERR1FO  054034
BIT8  = 000400 G        CONTIN  013414          C$MSG = 000023          DRVNAV  005430          ERR10   012734 G
BIT9  = 001000 G        COSTAT= 000040          C$PNTB= 000014          DSESTA= 000400          ERR2    011412 G
BLD.HW  046114          COUNT   002646          C$PNTF= 000017          DSMSK = 001400          ERR3    011460 G
```

```
ERR4    011526 G    FRMWD   007135      G$RADL= 000120      INITIA  057454      LPT.RE  045010
ERR5    011576 G    FWDSKO= 002000      G$RADO= 000020      INIT.M  046016      LSI.RE  045004
ERR6    011646 G    FWDSKS= 000400      G$RADT= 000100      INIT.R  040212 G    LUP     065536
ERR7    012524 G    F$AU  = 000015      G$XFER= 000004      INOUTS= 000020      LUP.AD  052762
ERR8    012574 G    F$BGN = 000040      G$YES = 000010      INPUTA  060402      L$AUT   002070 G
ERR9    012670 G    F$CLEA= 000007      HADONE  002422      INTEBL= 000100      L$CCP   002044 G
ESC.PC  052756      F$DU  = 000016      HCESTA= 040000      INTFOR  053700      L$CLEA  014200 G
EXACYL  002636      F$END = 000041      HCORED  045156      INTHLR  014332 G    L$DEPO  002011 G
EXHCYL  002634      F$HARD= 000004      HCOREQ  045036      INVAL.  045124      L$DEVP  002052 G
EXOCYL  002632      F$HW  = 000013      HCORET  040410 G    INVINT  053572      L$DISP  013200 G
EXROT   002640      F$INIT= 000006      HCRCER= 004000      INV.SW  041420      L$DR    002102 G
FBSFIL  003062      F$JMP = 000050      HDALIG= 000010      IN.SUF  046766      L$DRCT  002062 G
FILL    060274      F$MOD = 000000      HDCYL = 077600      I$AU  = 000041      L$DRS   002064 G
FILL.C  000204 G    F$MSG = 000011      HDHSEL= 000100      I$CLN = 000041      L$DRST  002102 G
FLAGS   040420 G    F$PWR = 000017      HDMOVF  006762      I$DU  = 000041      L$DTP   002040 G
FLAGTA  063446      F$RPT = 000012      HDRCMP= 000002      I$HRD = 000041      L$DU    014320 G
FLAG.I  045460      F$SEG = 000003      HDR40 = 100000      I$INIT= 000041      L$DUT   002072 G
FLA.SE  063414      F$SOFT= 000005      HDSEC = 000077      I$MOD = 000041      L$DVTY  002104 G
FLG.MA  045420      F$SRV = 000010      HDSEL = 000020      I$MSG = 000041      L$EF    002024 G
FMTOP1  010322      F$SUB = 000002      HDWD    007117      I$PWR = 000041      L$EXP1  002032 G
FMTOP2  010351      F$SW  = 000014      HDWRD1  002464      I$RPT = 000041      L$EXP2  002034 G
FMTOP3  010373      F$TEST= 000001      HDWRD2  002466      I$SEG = 000041      L$EXP3  002036 G
FMT1    010414      GARBAG  061000      HDWRD3  002470      I$SFT = 000041      L$HARD  037340 G
FMT1.1  010421      GETCHR  057504      HEAD  = 000006      I$SRV = 000041      L$HPCP  002046 G
FMT11   010640      GETCMN  063070      HEADLM= 010000      I$SUB = 000041      L$HPTP  002056 G
FMT12   010646      GETPAR  054562      HEADQ   040010      I$TST = 000041      L$HW    013146 G
FMT13   010654      GETPOS  020652      HEADV   040035      J$JMP = 000167      L$ICP   002042 G
FMT14   010720      GETSTA= 000003      HEADW   013170      KBPTR   040230 G    L$INIT  013246 G
FMT15   010752      GETSWI  062064      HERTZ.  044776      KBUF    040232 G    L$LADP  002076 G
FMT16   011006      GET.TW  061634      HFIN    002602      LABACF  006732      L$LAST  040176 G
FMT17   011017      GLBDAT  002112 G    HFINU   002604      LABACR  006746      L$MREV  002012 G
FMT18   011041      GLBEQA  002112 G    HFOUT   002606      LABEXP  006641      L$NAME  002000 G
FMT19   011073      GLBERR  011344 G    HFOUTU  002610      LABHCF  006702      L$REPP  002054 G
FMT2    010430      GLBSUB  014324 G    HICYL = 020000      LABHCR  006716      L$REV   002010 G
FMT20   011130      GLBTXT  004534 G    HILIM = 000004      LABIN   006616      L$SOFT  037464 G
FMT21   011160      GSTAT   014762      HILIMQ  037767      LABMID  006624      L$SPC   002030 G
FMT22   011203      GSTATC  014746      HILIMW  013166      LABOCF  006652      L$SPCP  002050 G
FMT23   011237      GSTATG  014772      HNFERR= 010000      LABOCR  006666      L$SPTP  002060 G
FMT24   011253      GSTATR  014732      HOLDSP= 000020      LABOUT  006633      L$STA   002066 G
FMT25   011260      GTSTAT= 000104      HOSTAT= 000020      LAB1    005547      L$SW    013162 G
FMT26   011270      G$EXCP= 000400      HPTCOD  013144 G    LAB2    005562      L$TIML  002022 G
FMT27   011314      G$HILI= 000002      HRDPRM  037336 G    LIMVAL  037753      L$TIMU  002020 G
FMT28   011333      G$LOLI= 000001      HRDWTS  024260 G    LINE.F  040452 G    L$TIM1  002016 G
FMT3    010433      G$NO  = 000000      HRIN    002612      LOAD.F  045414      L$TSTI  002074 G
FMT4    010436      G$OFFS= 000400      HRINU   002614      LOCERR  002656      L$UNIT  002014 G
FMT5    010447      G$OFSI= 000376      HROUT   002616      LOCYL = 040000      L.BA    002450
FMT6    010467      G$PRMA= 000001      HROUTJ  002620      LOGMSG  057446      L.CLK.  044762
FMT7    010531      G$PRMD= 000002      HSMSK = 000100      LOLIM = 000002      L.CS    002446
FMT8    010601      G$PRML= 000000      HSSTAT= 000100      LOLIMQ  037732      L.DA    002452
FMT9    010633      G$RADA= 000140      H$AAB   064236      LOLIMW  013164      L.MP    002454
FOLWRT= 000100      G$RADB= 000000      IBUFF   003256      LPBFR   040226 G    L10000  011410
FORM.T  054044      G$RADD= 000040      ININIT  040400 G    LPCNTR  040224 G    L10001  011456
FREE    061352      G$RADF= 000200      INITCO  013246 G    LPT.AD  045014      L10002  011524
```

```
L10003  011574          MANQ    037670          MQUALS= 003760          OPFLAG  002416          PRIO7 = 000340 G
L10004  011644          MAN.TI  042504          MREAD   004542          OPIERR= 002000          PRNTST  060644
L10005  012522          MAPROX  006606          MREADH  004555          OPMSGS  002112          PRO.CM  045366
L10006  012572          MAP16   070500 G        MRESKO  005216          OPRO04  007067          PSETNM  002654
L10007  012666          MASK.B  047530          MREVSK  005074          OPR1A   007040          PTAB.S  040406 G
L10010  012732          MASK.W  047526          MRLFAL  010036          OPR1B   007044          PUTCHR  057460
L10011  013142          MBADAD  005252          MRSLT   004747          OPR12   007021          PWCON   013540
L10012  013160          MBADSF  005273          MSEEK   004534          ORIN    002566          PWRFLG  002662
L10013  013176          MBSETO= 000001          MSPERR  007516          ORINU   002570          PWR.FA  070734 G
L10014  014176          MCERR   007344          MSTERR  007551          ORMID   002572          PWR.FL  040210 G
L10015  014316          MCONHN  005705          MTMBS   005350          ORMIDU  002574          PWR.MS  071062
L10016  014322          MCYLOC  007724          MTOSLO  005603          OROUT   002576          PWR.SA  071056
L10017  014366          MCYLUP  004774          MUL     067776 G        OROUTU  002600          PWR.UP  071060
L10020  024600          MDATCP  004636          MULOAD  004763          OUTINS= 000040          P.CLK.  044770
L10021  024544          MDCRC   007366          MUNDEF  007771          O$APTS= 000000          P2T01E  006015
L10022  025070          MDHEDR  002000 G        MWDERR  007605          O$AU  = 000000          P2T02E  006015
L10023  025034          MDLT    007413          MWGERR  007534          O$BGNR= 000000          P2T03E  006034
L10024  025302          MDRDY   007333          MWORD   005575          O$BGNS= 000001          P2T04E  006060
L10025  025264          MDRERR  007455          MWRCHK  004572          O$DU  = 000001          P2T05E  006102
L10026  025512          MDRRES  005622          MWRITE  004606          O$GNSW= 000001          P2T06E  006124
L10027  025510          MDRVST  007503          MWRSET  004717          O$POIN= 000001          P2T07E  006146
L10030  025722          MDSERR  007466          MWRTAB  010077          PARSES  063142          P2T08E  006172
L10031  025720          MEM.SI  045024          M40HDR  004703          PART2 = 000000          P2T09E  006214
L10032  026146          MERRS   010160          NEWCYL  002514          PAR.LA  057134          P2T10E  006221
L10033  026144          MEXERS  010115          NEWPRI  067522 G        PASCNT  002644          P2T11E  006236
L10034  026346          MFLERR  007656          NEXTAR  063632          PASNEW  013442          P2T12E  006253
L10035  026330          MFMTER  005323          NOCLR = 000010          PASNUM  002652          P2T13E  006267
L10036  026574          MFOLWR  005050          NOERCT  002657          PATTBL  002224          P2T14E  006307
L10037  026572          MFWDSK  005127          NOIRPT= 000002          PAT1    004256          P2T15E  006332
L10040  027070          MFWSKO  005162          NOOP  = 000100          PAT10   004532          P2T16E  006362
L10041  027020          MGTSTA  004622          NOPWR   005463          PAT2    004260          P2T17E  006405
L10042  027366          MHCERR  007567          NO.CLK  042534          PAT3    004320          P2T18E  006445
L10043  027330          MHCRC   007356          NO.FLA  063426          PAT4    004360          P2T19E  006472
L10044  027664          MHDERR  007641          NO.LPT  060744          PAT5    004420          RDALHD  021120
L10045  027632          MHDRCP  004661          NO.PTA  045224          PAT6    004426          RDDATA= 000114
L10046  031432          MHFCRC  007425          NR    = 000000          PAT7    004466          RDHEAD= 000110
L10047  032142          MHNF    007377          NUMBIN  054070          PAT8    004470          RDNOHR= 000116
L10050  032356          MINOUT  005025          NUM.LA  054236          PAT9    004530          RDYCHK  016664
L10051  032302          MIN.IN  040176 G        NUM.UN  040574          PH65$   016626          RDYWAI  020404
L10052  033102          MIN.US  040200 G        NUNITS  047504          POSHDS  016324          READRL  014530
L10053  034124          MISWI = 000000          NXMERR= 020000          POSHDO  020360          READ.P  065640 G
L10054  034044          MISWIN  013162          NXTFOR  063702          POSHSB  020354          REGBAC  070464 G
L10055  035116          MITEST= 100000          NXTPAS  013434          POSHW1  020346          REGSAV  070450 G
L10056  034776          MNDRST  007731          OBUFF   003656          PRINTC  060754          RELDWT= 040000
L10057  036226          MNEERR  007704          OCTMSG  057432          PRINTF  064256          REQN.P  045376
L10060  036146          MNOCLR  005731          OFIN    002552          PRIOR = 000004          REQN.T  045370
L10061  037334          MNOINT  005652          OFINU   002554          PRIO0 = 000000 G        RESE3   010173
L10062  037254          MODR    070042 G        OFMID   002556          PRIO1 = 000040 G        RESE4   010177
L10063  037412          MOPER   004733          OFMIDU  002560          PRIO2 = 000100 G        RESE5   010204
L10064  037626          MOPERR  007622          OFOUT   002562          PRIO3 = 000140 G        RESE6   010211
MAJ.IN  040202 G        MORECE  002426          OFOUTU  002564          PRIO4 = 000200 G        RESPAR  002474
MAJ.LO  065636          MOUTIN  005004          OLDCYL  002512          PRIO5 = 000240 G        RESTAR  013404
MAJ.US  040204 G        MPNAM   005542          ONSWAP  017210          PRIO6 = 000300 G        RESTBL  002164
```

OUTERR MACY11 30(1046) 04-NOV-77 13:30 PAGE 87-3
DZRLDA.SUP 11-OCT-77 15:40 SYMBOL TABLE SEQ 0138

```
REVSKO= 001000
REVSKS= 000200
RE.SET  041566
RLBA  = 000002
RLBAS   002440
RLCS  = 000000
RLCSR = 000000
RLDA  = 000004
RLDRV   002444
RLMP  = 000006
RLVEC   002442
RORWOP= 020000
RPTOP   023000
RPTREM  023774
RPTRES  023566
RSTACK  067724 G
RSTRT   013330
RSX.FL  045410
SAMSK = 000077
SBSFIL  002666
SEARCH  061602
SECQ    037650
SECWD   007123
SEEK  = 000106
SEEKOP= 010000
SEGSTA  040424 G
SEQMES  007156
SETDON  013462
SET.MA  045604
SFTPRM  037462 G
SGNWD   007112
SHIFT   070562 G
SIZE.C  067440 G
SIZE.M  067356 G
SIZ.TR  067516
SKTMES  006504
SPDSTA= 004000
SPEC.U  045314
SPTCOD  013160 G
SPV.SE  042016
SRTMES  006520
SSINDX  002414
STAMES  007221
STAMSK= 000007
STARTC  067306 G
STATE2  010216
STATE3  010226
STATE5  010236
STOSTA= 010000
STRCHR  060334
STREQ.  045136
STRT.T  045372
ST.REQ  045312
ST.SET  041636
SUBSTK  002250
SUNIT.  045400
SUPERV  043356
SUPFLA  040364 G
SUPV.T  040542 G
SUP.PR  041410
SVCBGL= 000001
SVCCNT= 177777
SVCGBL= 177777
SVCHAN  047710
SVCINS= 000000
SVCSTK= 177777
SVCSUB= 000001
SVCTAG= 000000
SVCTST= 000001
SWAPHD  017150
SWCHAN  045216
SWITCH  063606
SW.PTA  045202
SYS.FT  053522
S$LSYM= 010000
TBLSTR  002436
TCERR   007310
TEMP0   002530
TEMP1   002532
TEMP2   002534
TEMP3   002536
TEMP4   002540
TEMP5   002542
TEMP6   002544
TEMP7   002546
TEMP8   002550
TERMI   065626
TERMLI  063434
TERMTA  057416
TEST.M  045326
TIMFLG  040354 G
TIM.CO  040206 G
TIM.OP  054042
TOO.MA  057376
TOSLOW= 000001
TRPFLG  002660
TRPHAN  014324
TSTINT  014714
TSTLAB  006007
TST.AB  047642
TST.TO  041446
TYPEC   057772
TYPEPC  053666
TYPFLA  063310
TYPLIN  057670
TYPNUM  057256
TYPSTR  057710
TYP.ER  053552
TY.UNI  047006
T$ARGC= 000007
T$CODE= 000130
T$ERCO= 000043
T$ERRN= 003247
T$EXCP= 000000
T$FLAG= 000040
T$HILI= 000377
T$LOLI= 000001
T$LSYM= 010000
T$MCAL= 177777
T$NEST= 177777
T$NSK0= 000000
T$NSK1= 000005
T$NSK2= 000002
T$SAVL= 177777
T$SEGL= 177777
T$SEK0= 010000
T$SUBN= 000001
T$TAGL= 177777
T$TAGN= 010065
T$TEMP= 000000
T$TEST= 000023
T$TSTM= 177777
T$TSTS= 000001
T$$CLE= 010015
T$$DU = 010016
T$$HAR= 010063
T$$HW = 010012
T$$INI= 010014
T$$MSG= 010011
T$$SEG= 010000
T$$SOF= 010064
T$$SRV= 010017
T$$SUB= 010062
T$$SW = 010013
T$$TES= 010061
T.BA    002460
T.CS    002456
T.DA    002462
T.MP    002464
T.STAT  002472
T1      024260 G
T1.1    024330
T10     027072 G
T10.1   027212
T11     027370 G
T11.1   027512
T12     027666 G
T13     031434 G
T14     032144 G
T14.1   032242
T15     032360 G
T16     033104 G
T16.1   033364
T17     034126 G
T17.1   034146
T172$   024330
T1765$  024600
T18     035120 G
T18.1   035454
T1865$  025070
T187$   024656
T19     036230 G
T19.1   036532
T1965$  025302
T197$   025116
T2      024602 G
T2.1    024656
T206$   025356
T2065$  025512
T216$   025566
T2165$  025722
T2265$  026146
T227$   026014
T233$   026174
T2365$  026346
T2465$  026574
T247$   026440
T25TBL  002274
T2517$  027060
T256$   026634
T2565$  027070
T257$   026672
T258$   026654
T266$   027116
T2665$  027366
T267$   027200
T275$   027414
T276$   027500
T2765$  027664
T3      025072 G
T3.1    025126
T306$   032210
T3065$  032356
T307$   032242
T310$   032250
T3100$  033224
T3101$  033364
T3165$  034124
T3204$  035000
T3265$  035116
T33TBL  002322
T3300$  035262
T3301$  035454
T3365$  036226
T3400$  036356
T3401$  036532
T3465$  037334
T4      025304 G
T4.1    025354
T5      025514 G
T5.1    025564
T6      025724 G
T6.1    026012
T7      026150 G
T7.1    026206
T8      026350 G
T8.1    026436
T9      026576 G
T9.1    026734
ULOAD = 000010
UNDTST  007054
UNI.MA  045316
UNXERR  005772
USER.P  040402 G
USER.T  040404 G
VALDES  006547
VALID.  040644
VAL.LA  041374
VAL.SW  045432
VCNRST  005751
VCSTAT= 001000
VECMSG  037426
VECT  = 000002
VERHDR  020016
VERPOS  021000
WAITIN  014562
WCMSK = 017777
WCRNG = 160000
WDESTA= 100000
WGESTA= 002000
WIDTH   054436
WLSTAT= 020000
WRTSWI  002434
WTDATA= 000112
XEQDIA  067610 G
XEQSUB  067576 G
XEQ.CL  047446
XEQ.CM  044754
XEQ.IN  047130
XEQ.LA  043312
XEQ.OP  047222
XEQ.PR  042544
XEQ.TE  047266
XRDHD   017376
XRDHDC  017366
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:30  PAGE 87-4
DZRLDA.SUP    11-OCT-77 15:40          SYMBOL TABLE                                                    SEQ 0139

XRDHDG  017402            XSEEK1  015630          XWRITT  022100          X$OFFS= 000400          $SAV2   070626 G
XREAD   022150            XTIME   066316 G        XWRIT1  022114          X$TRUE= 000020          $SAV3   070642 G
XREADG  022156            XTIMEN  067142          XXDP.D  045164          X1X1   = 072000          $SAV4   070660 G
XSEEK   015624            XTIMST  066340          X$ALWA= 000000          $BREG   045472          $SAV5   070700 G
XSEEKT  015614            XWRITE  022110          X$FALS= 000040          $ENDAD  067562 G          .      = 072000


. ABS.   072000     000


 ERRORS DETECTED:   0

 DSKZ:DZRLDA,DSKZ:DZRLDA/EQ:PART2=DZRLDA.SML,DZRLDA.PT1,DZRLDA.P11,DZRLDA.PT2,DZRLDA.SUP
 RUN-TIME:  44 49 1 SECONDS
 RUN-TIME RATIO:  1434/94=15.1
 CORE USED:   19K  (37 PAGES)
```