RL11,RLV11,RL01

RL01 DRIVE TEST PART 1

MD-11-DZRLC-A

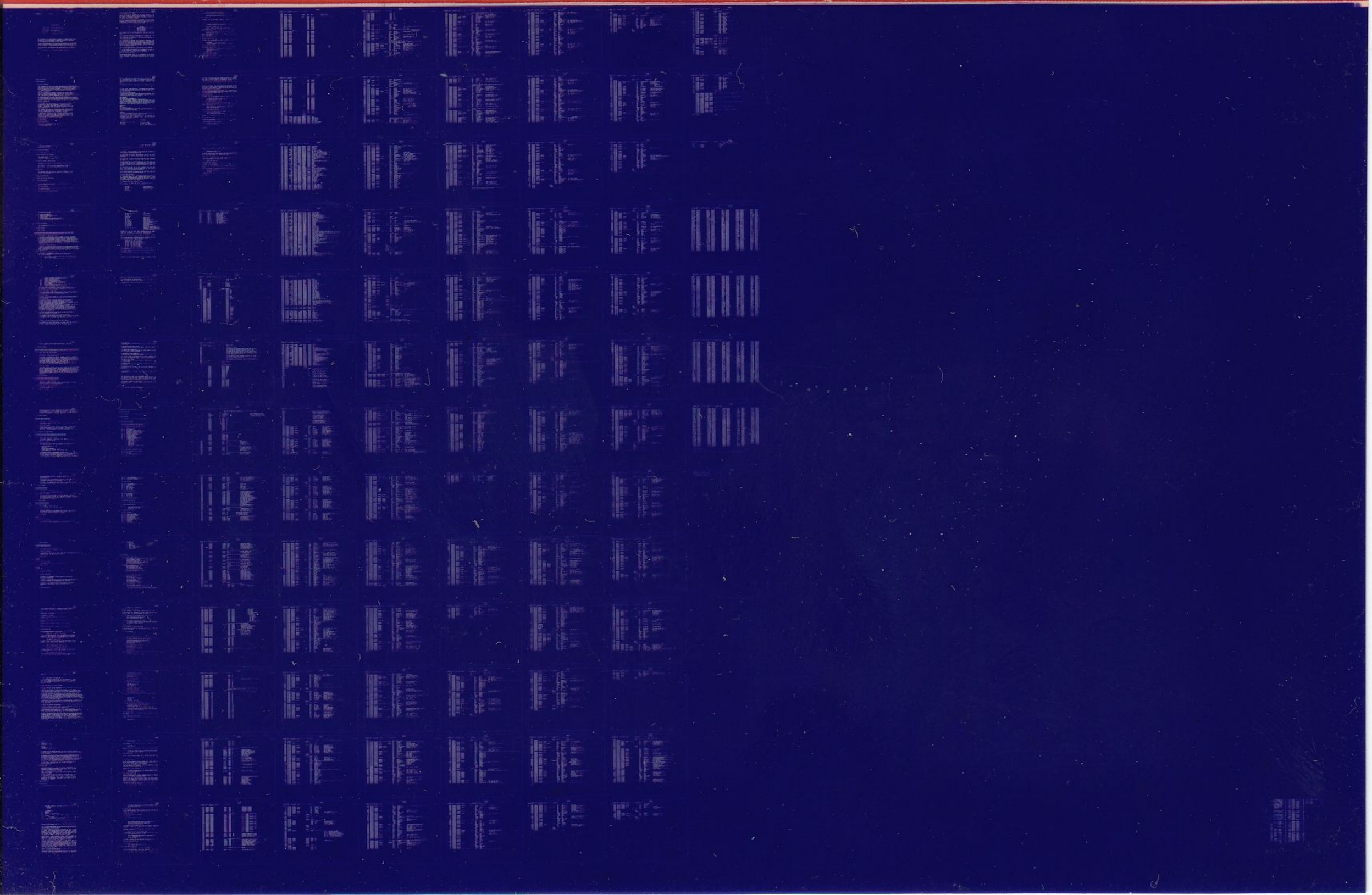
EP-DZRLC-A-DL

DEC 1977

FICHE 1 OF 1

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRLC-A-D
PRODUCT NAME: RL01 DRIVE TEST (PART 1)
DATE CREATED: 28 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 TESTS THE RL01 INTERFACE AND BASIC DRIVE LOGIC. GET STATUS WITH RESET, GET STATUS, SEEK, AND READ HEADER ARE THE ONLY COMMANDS EXECUTED IN THE PROGRAM. ONLY SEEKS WITH 0 DIFFERENCE ARE USED SO NO HEAD MOVEMENT IS REQUIRED.

A SIGNIFICANT PORTION OF THE PROGRAM REQUIRES MANUAL INTERVENTION. THESE TESTS TEST THE COVER OPEN AND WRITE LOCK STATUS. THE DRIVE MUST BE LOADED AND UNLOADED TO TEST ALL THE CONDITIONS OF HEADS OUT, BRUSH HOME, AND DRIVE STATES. THE PROGRAM CAN BE RUN IN AUTOMATIC MODE IN WHICH CASE ALL TESTS REQUIRING MANUAL INTERVENTION ARE BYPASSED.

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 RLO1K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

MAINDEC-11-DZRLC-A

1.3 RELATED DOCUMENTS AND STANDARDS

RLO1 USERS MANUAL (EK-RLO1-UG-PRE)
XXDP USERS MANUAL
DIAGNOSTIC SUPERVISOR PROGRAM LISTING

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RLO1 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

MD-11-DZRLA	RL11/RLV11 RLO1 CONTROLLER TEST (PART 1)
MD-11-DZRLB	RL11/RLV11 RLO1 CONTROLLER TEST (PART 2)
MD-11-DVRLA	RLV11 RLO1 DISKLESS TEST (RLV11 ONLY)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RLO1 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 SOFTWARE 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, IN-
STEAD 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 RE-
QUESTED 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 UN-
SATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE RE-
MAINS, 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.

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) O?

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.

EXECUTE DRIVE SELECT TESTS (N)?

IF "YES" TESTS 5 AND 6 ARE EXECUTED IN THE FIRST PASS OF THE PROGRAM. THESE TESTS REQUIRE MANUAL INTERVENTION TO CHANGE ADDRESS PLUGS AND REQUIRE A FULL COMPLEMENT OF ADDRESS PLUGS (O - 3).

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

IF "YES", TEST 11 IS EXECUTED IN THE FIRST PASS.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF "YES", TESTS 1, 2, 3, AND 4 ARE EXECUTED TO TEST BASIC INTTERFACE OPERATIONS, HEAD LOADING, HEAD UNLOADING, AND ALL STATE CHANGES.

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.

SEQ 0011

DROP DRIVE IF NO RESPONSE (N)?

IF THIS PARAMETER IS SPECIFIED AS YES, THE PROGRAM WILL CHECK EACH DRIVE BEFORE TESTING STARTS TO DETERMINE IF IT IS READY OR IF IT WILL RESPOND TO A GET STATUS. 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 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 AT LEAST 3 MINUTES. THE ACTUAL AMOUNT OF TIME REQUIRED WILL DEPEND ON THE SPEED WITH WHICH THE REQUESTED OPERATIONS ARE PERFORMED BY THE USER.

FIRST PASS EXECUTION ON MORE THAN ONE DRIVE WITH DRIVE SELECT TESTS WILL REQUIRE AT LEAST (TBD). ACTUAL TIME WILL AGAIN DEPEND ON THE USER.

IF HEAD ALIGNMENT SUPPORT IS REQUESTED, THE PROGRAM RUN TIME IS CONTROLED BY THE DURATION OF THE HEAD ALIGNMENT OPERATION.

IF MANUAL INTERVENTION, DRIVE SELECT, OR HEAD ALIGNMENT SUPPORT ARE NOT REQUESTED, (OR IN SECOND AND SUBSEQUENT PASSES REGARDLESS OF REQUESTED OPERATION) THE RUN TIME IS APPROXIMATELY 3 SECONDS.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

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)                        SEQ 0013
         (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.

3.1.1 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

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 HDS 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.

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 BASIC INTERFACE TEST (PART 1)

LOAD IN DRIVE NUMBER. DO GET STATUS WITH RESET. IF OPI SETS:
DRIVE INTERFACE IS DEAD
DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
MARKER DETECTION FAILED
DRIVE IS NOT SELECTING OR AC LOW IS SET
SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT WITH NO OPI, CHECK STATUS RECEIVED. COVER OPEN AND BRUSH HOME SHOULD BE SET. IF NOT:
BAD STATUS DATA LINE
BAD COVER SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER
BAD BRUSH HOME SWITCH OR LOGIC

CHECK WRITE LOCK STATUS BIT SET. IF NOT:
BAD SWITCH OR WRITE LOCK LOGIC

DRIVE COMMAND SHIFT REGISTER

CHECK STATE FOR 0. IF NOT:
BAD STATE ROM
DRIVE COMMAND SHIFT REGISTER

CHECK VOLUME CHECK RESET. IF NOT:
BAD RESET DETECTION
BAD VOLUME CHECK LOGIC
DRIVE COMMAND SHIFT REGISTER

CHECK DRIVE ERROR RESET. IF NOT:

BAD DRIVE ERROR INTERFACE
SOME OTHER ERROR STUCK ON. REPORT WHICH ERROR.

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2 IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED,

AND IS RUN IN FIRST PASS ONLY.

TEST 2 BASIC INTERFACE TEST (PART 2)

REQUEST OPERATOR TO CLOSE COVER AND RESET WRITE LOCK.

DO GET STATUS LOOP CHECKING IF COVER OPEN OR WRITE LOCK RESETS. WAIT 15 SECONDS FOR BOTH TO CHANGE. IF NO CHANGE, ASK OPERATOR TO TYPE CR IF PROCEDURE WAS FOLLOWED.

IF ONE CHANGED BUT NOT THE OTHER, REPORT WHICH FAILURE:

WRITE LOCK SWITCH OR LOGIC
(OR) COVER OPEN SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER

IF NEITHER CHANGED, REPORT BOTH FAILURES.

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

TEST 3 HEAD LOADING TEST

REQUEST OPERATOR TO PRESS LOAD SWITCH.

DO GET STATUS LOOP CHECKING FOR STATE TO GO TO 1. WAIT 30 SECONDS FOR CHANGE. IF NO CHANGE, ASK OPERATOR TO CONFIRM ACTION BY TYPING CR.

IF LOAD WAS PRESSED:

BAD STATE ROM

BAD LOAD SWITCH OR LOGIC

CHECK THAT STATE 1 REMAINS FOR LESS THAN 30 SECONDS. IF NOT:

SPINDLE NOT TURNING OR TO SLOW (AC SERVO)
SECTOR PULSE DETECTION OR LOGIC BAD
BAD CLOCK SHIFT REGISTER IN SPEED CONTROL
BAD DISK ON SPEED LOGIC
BAD STATE ROM

AND CHECK IF SPINUP TIMEOUT ERROR SET. IF NOT:

BAD STATE ROM
BAD TIMEOUT DETECTION LOGIC

CHECK THAT STATE GOES TO 2. IF NOT:

BAD STATE ROM

CHECK THAT BRUSH HOME IS RESET 5 SECONDS OR LESS AFTER STATE IS 2. IF NOT:

BAD BRUSH HOME SWITCH OR LOGIC

BAD BRUSH MOTOR (AC SERVO)

WAIT 30 SECONDS FOR BRUSH HOME TO SET. IF NOT:

BAD AC SERVO
BAD SWITCH OR LATCH

CHECK THAT STATE HAS CHANGED TO 3. IF NOT:

BAD STATE ROM

AFTER STATE IS 3, CHECK HEADS OUT IS SET. IF NOT:

BAD SWITCH
BAD SEEK CONTROL ROM
BAD VELOCITY ROM
BAD DC SERVO

CHECK VOLUME CHECK IS SET. IF NOT:

BAD VOLUME CHECK LOGIC

CHECK IF DRIVE ERROR IS SET. IF NOT:

BAD DRIVE ERROR LOGIC OR INTERFACE

WAIT 300 MS FOR STATE TO CHANGE TO 4. IF IT DOESN'T CHANGE:

STATE ROM BAD
SEEK ROM
VEL ROM
GUARD BAND DETECTION

WAIT 15 MS FOR STATE TO CHANGE TO 5.

8 MS AFTER STATE GOES TO 5, DRIVE READY SHOULD SET. IF NOT:

INTEGRATOR OR NULL DETECTION FAILURE
READY ONE SHOT BAD
ENABLE TIMEOUT H NOT SETTING OR COUNT LOGIC BAD

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

TEST 4 HEAD UNLOADING TEST

CHECK DRIVE IS READY. IF NOT REPORT AND ASK OPERATOR TO MAKE DRIVE READY.

REQUEST OPERATOR TO UNLOAD DRIVE.

LOOP ON GET STATUS WAITING FOR STATE TO CHANGE TO 6. IF NO CHANGE:

BAD STATE ROM
BAD SWITCH

WAIT 300 MS FOR STATE TO CHANGE TO 7. IF NO CHANGE:

BAD STATE ROM

AFTER STATE IS 7, WAIT 30 SEC FOR STATE TO CHANGE TO STATE 0. IF NO CHANGE:

NO BRAKING
BAD AC SERVO

REQUEST OPERATOR TO LOAD DRIVE. WAIT UNTIL DRIVE BECOMES READY.

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

TEST 5 DRIVE SELECT TEST

INSTRUCT THE OPERATOR TO REMOVE DRIVE ADDRESS PLUGS FROM ALL DRIVES EXCEPT THE DRIVE UNDER TEST. ASK THAT CARRIAGE RETURN BE TYPED WHEN DONE.

DO GET STATUS TO ADDRESS OF DRIVE UNDER TEST. CHECK THAT NO ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER ADDRESSES AND CHECK THAT OPI SETS FOR ALL OTHER ADDRESSES.

DO GET STATUS TO ADDRESS OF NEXT SEQUENTIAL ADDRESS. CHECK THAT NO ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER ADDRRESSES AND CHECK THAT OPI SETS.

REPEAT FOR ALL DRIVE ADDRESSES (0,1,2,3 - 0 IS SEQUENTIAL AFTER 3).

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

TEST 6 DRIVE SELECT ERROR TEST

REQUEST OPERATOR INSERT IDENTICAL ADDRESS PLUGS IN TWO DRIVES (MUST BE IDENTICAL TO NUMBER SPECIFIED EARLIER). REQUEST OPERATOR TYPE CARRIAGE RETURN WHEN READY.

PROCEDURE WILL BE TO GET STATUS AND CHECK FOR DRIVE SELECT ERROR. THEN RESET THAT DRIVE AND VERIFY THAT DRIVE SELECT ERROR IS NOT REPORTED AGAIN. WAIT 1 SECOND, THEN CHANGE DRIVE SELECT TO A DIFFERENT NUMBER AND BACK AGAIN. DRIVE SELECT ERROR SHOULD SET AGAIN.

OPERATOR SHOULD SEE THE FAULT LIGHT ON ON BOTH DRIVES. IF INDICATOR IS NOT SEEN ON A DRIVE:

DRIVE SELECT ERROR DETECTION IS BAD IN THAT DRIVE.

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

4.2 STANDARD TESTS

IF THE PROGRAM OPERATION MODE 1 IS SELECTED, THIS WILL BE THE FIRST TEST EXECUTED. THE DRIVE(S) TO BE TESTED MUST BE POWERED UP, HEADS LOADED, AND WRITE LOCK RESET.

TEST 7 INITIAL STATE TEST

DO GET STATUS, WAIT FOR INTERRUPT.

IF OPI OCCURS:

DRIVE INTERFACE IS DEAD

DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
DRIVE IS NOT SELECTING OR AC LOW IS SET
SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT OCCURS WITHOUT OPI, CHECK DRIVE READY. READY SET INDICATES HEADS ARE LOADED AND ARE TRACKING (POSITION WORKING).

IF MANUAL INTERVENTION TESTS WERE RUN, CHECK THAT HEAD 0 IS SELECTED. IF NOT:

DRIVE CYCLE UP DID NOT SELECT HEAD 0

IF DRIVE READY IS SET, CHECK STATUS MESSAGE RECEIVED. HEADS OUT AND BRUSH HOME MUST BE SET. IF NOT:

DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
HEADS OUT OR BRUSH HOME SWITCH OR ASSOCIATED
CIRCUITRY BAD
STATUS DATA BAD

IF MANUAL INTERVENTION TESTS WERE RUN AND THIS IS THE FIRST PASS CHECK THAT VOLUME CHECK AND DRIVE ERROR ARE SET.

CHECK ALL ERROR BITS ARE 0.

CHECK STATE IS 5. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD

TEST 8 INITIAL RESET STATE TEST

DO GET STATUS HEAD SELECT = 0, WAIT FOR INTERRUPT.

DO GET STATUS WITH RESET, WAIT FOR INTERRUPT. BOTH DRIVE ERROR AND VOLUME CHECK SHOULD NOW BE RESET. IF NOT:

RESET DETECTION, RESET ERROR, OR VOLUME CHECK FLOP BAD

DRIVE COMMAND SHIFT REGISTER BAD

HEAD SELECTED BIT SHOULD STILL BE ZERO. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
HEAD SELECT SHIFT REGISTER NOT LOADING

TEST 9 DRIVE READY TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. WAIT FOR INTERRUPT. GET STATUS. CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

CHECK DRIVE READY IS RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS MAY HAVE MOVED (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED

CHECK DRIVE ERROR DID NOT SET. IF IT SET, DO GET STATUS AND REPORT WHICH ERROR.

VERIFY HEAD SELECT IS ZERO.

TEST 10 SEEK SIGN SWITCH TEST

DO SEEK WITH DIFFERENCE 0, SIGN 1, HEAD 0. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

COUNT ROM
DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

VERIFY DRIVE IS NOT READY

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
COUNT ROM

VERIFY DRIVE ERROR DID NOT SET

VERIFY HEAD SELECT IS ZERO.

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

TEST 11 HEAD ALIGNMENT SUPPORT ROUTINE

THIS TEST IS EXECUTED WHEN THE PROGRAM IS STARTED AT ADDRESS 204, HEAD ALIGNMENT SUPPORT IS REQUESTED, AND IN THE FIRST PASS ONLY. IT IS BYPASSED IF THE PROGRAM IS STARTED AT ANY OTHER ADDRESS AND IN THE SECOND AND SUBSEQUENT PASSES.

THIS TEST SELECTS THE DRIVE UNDER TEST AND LOOPS ON A GET STATUS WITH RESET. THE WRITE LOCK BIT IS MONITORED AND WHEN WRITE LOCK IS RESET HEAD 0 IS SELECTED AND WHEN WRITE LOCK IS SET HEAD 1 IS SELECTED. THIS WILL PERMIT THE HEADS TO BE ALIGNED IN KEEPING WITH THE PRESENT HEAD ALIGNMENT PROCEDURE WITHOUT RETURNING TO THE CONSOLE.

TYPING A CARRIAGE RETURN ON THE CONSOLE WILL TERMINATE THIS TEST ON THE DRIVE UNDER TEST. BEFORE TERMINATING, THE TEST WILL CHECK THAT WRITE LOCK IS RESET. IF NOT, THE OPERATOR WILL BE REQUESTED TO RESET WRITE LOCK.

TEST 12 HEAD SWITCHING TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 1. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER IS PICKING UP BITS
ASSOCIATED CIRCUITRY IS BAD

VERIFY DRIVE READY RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
DRIVE CANNOT TRACK WITH THIS HEAD

VERIFY DRIVE ERROR DID NOT SET.

DO GET STATUS, CHECK HEAD SELECT IS CORRECT. IF NOT:

HEAD SELECT REGISTER BAD
DRIVE COMMAND SHIFT REGISTER BAD

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. REPEAT ABOVE TESTS.

TEST 13 READ HEADER TEST (PART 1)

DO SEEK WITH DIFFERENCE 0, HEAD 0, SIGN 0. WAIT FOR INTERRUPT AND WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT.

CHECK IF HEADER CRC ERROR SET. IF SET:

READ/WRITE BOARD BAD

READ DATA LINE BAD

CHECK IF BIT 6 OF WORD 1 IS SAME AS HEAD SELECT BIT IN STATUS. IF NOT:

HEADS ARE SWITCHED (CABLE)
HEAD SELECT LOGIC

IF MANUAL INTERVENTION TESTS WERE RUN AND HEAD ALIGNMENT TESTS WERE NOT RUN, CHECK THAT HEADER WORD 0 INDICATES HEADS ARE POSITIONED OVER CYLINDER 0. STORE HEADER WORD 1.

REPEAT TESTS USING HEAD 1.

CHECK THAT CYLINDER PORTION OF STORED HEADER WORD 1 IS THE SAME AS HEADER WORD 1 OF THIS HEADER. IF NOT:

HEADS ARE MISALIGNED

TEST 14 READ HEADER TEST (PART 2)

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 0. WAIT FOR INTERRUPT. WAIT FOR READY.

DO 40 CONSECUTIVE READ HEADER, STORE 3 HEADER WORDS AFTER EACH READ.

CHECK ALL HEADERS FOR SEQUENCE AND CONTENT (WORD 2 ALL ZERO, BIT 15 WORD 1 AND 3 IS 0, HS BIT WORD 1 IS 0). IF NOT:

BAD READ/WRITE BOARD
BAD PACK

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 1. REPEAT ABOVE TEST FOR HEAD 1. %

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  DZRLC,A,0
(5) 002000      104                     .ASCII  @D@
(5) 002001      132                     .ASCII  @Z@
(5) 002002      122                     .ASCII  @R@
(5) 002003      114                     .ASCII  @L@
(5) 002004      103                     .ASCII  @C@
(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  014464                      .WORD   L$DISPATCH
(4) 002042  014520                      .WORD   L$INIT
(4) 002044  015466                      .WORD   L$CLEAN
(4) 002046  031404                      .WORD   L$HARD
(4) 002050  031530                      .WORD   L$SOFT
(4) 002052  002104                      .WORD   L$DVTYP
(4) 002054  000000                      .WORD   0
(4) 002056  014432                      .WORD   L$HW
(4) 002060  014446                      .WORD   L$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  015606                      .WORD   L$DU
(4) 002074  000000                      .WORD   0
(4) 002076  032004                      .WORD   L$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
02850                           ;       OFFSETS FOR HARDWARE P-TABLE
02851   000000                  CSR     =0                              ;BUS ADDRESS
02852   000002                  VECT    =2                              ;VECTOR ADDRESS
02853   000004                  PRIOR   =4                              ;PRIORITY
02854   000006                  DRSB    =6                              ;DRIVE SELECT BIT
02855   000010                  CNT     =10                             ;CONTROLLER TYPE
02856
02857                           ;       OFFSET FOR SOFTWARE P-TABLE
02858   000000                  MISWI   =0                              ;SOFTWARE PARAMETERS SWITCHES
02859   000002                  LOLIM   =2                              ;CYLINDER LOWER LIMIT
02860   000004                  HILIM   =4                              ;CYLINDER HIGH LIMIT
02861   000006                  HEAD    =6                              ;SELECTED HEAD FOR RUNNING TESTS
02862   000010                  ERLIM   =10                             ;ERROR LIMIT
02863   000012                  DCLIM   =12                             ;DATA COMPARE ERROR LIMIT
02864
02865                           ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
02866   000001                  ALLCYL  =BIT00                          ;USE ALL CYLINDERS
02867   000002                  ALLSEC  =BIT01                          ;USE ALL SECTORS
02868   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
```

```
                    004000          DCKERR  =4000                   ;DATA CHECK ERROR
                    004000          HCRCERR =4000                   ;HEADER CHECK ERROR
                    002000          OPIERR  =2000                   ;OPERATION INCOMPLETE ERROR
                    001400          DSMSK   =1400                   ;DRIVE SELECT MASK
                    000200          CRDYMSK =200                    ;CONTROLLER READY MASK
                    000100          INTEBL  =100                    ;INTERRUPT ENABLE MASK
                    000060          BAMSK   =60                     ;BUS ADDRESS UPPER MASK
                    000001          DRDYMSK =1                      ;DRIVE READY MASK

                                    ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
                    000077          SAMSK   =77                     ;SECTOR ADDRESS MASK
                    000100          HSMSK   =100                    ;HEAD SELECT MASK
                    077600          CAMSK   =77600                  ;CYLINDER ADDRESS MASK

                                    ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
                    000001          MBSET0  =1                      ;MUST BE SET, BIT 0
                    000004          DIRBIT  =4                      ;DIRECTION BIT
                    000020          HDSEL   =20                     ;HEAD SELECT BIT
                    077600          DIRMSK  =77600                  ;CYLINDER DIFFERENCE MASK

                                    ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
                    000003          GETSTAT =3                      ;GET STATUS SETUP
                    000010          DRSET   =10                     ;DRIVE RESET MASK

                                    ;       REGISTER BIT DEFINITIONS - MP FOR DATA XFER
                    017777          WCMSK   =17777                  ;WORD COUNT MASK
                    160000          WCRNG   =160000                 ;WORD COUNT RANGE MASK

                                    ;       REGISTER BIT DEFINITIONS - MP FOR READ HEADER
                    077600          HDCYL   =077600                 ;CYLINDER MASK
                    000077          HDSEC   =77                     ;SECTOR MASK
                    000100          HDHSEL  =100                    ;HEAD SELECT MASK

                                    ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
                    000007          STAMSK  =7                      ;STATE MASK
                    000010          BHSTAT  =10                     ;BRUSH HOME STATUS
                    000020          HOSTAT  =20                     ;HEADS OUT STATUS
                    000040          COSTAT  =40                     ;COVER OPEN STATUS
                    000100          HSSTAT  =100                    ;HEAD SELECT STATUS
                    000400          DSESTAT =400                    ;DRIVE SELECT ERROR STATUS
                    001000          VCSTAT  =1000                   ;VOLUME CHECK STATUS
                    002000          WGESTAT =2000                   ;WRITE GATE ERROR STATUS
                    004000          SPDSTAT =4000                   ;SPIN ERROR STATUS
                    010000          STOSTAT =10000                  ;SEEK TIMEOUT ERROR STATUS
                    020000          WLSTAT  =20000                  ;WRITE LOCK STATUS
                    040000          HCESTAT =40000                  ;HEAD CURRENT ERROR STATUS
                    100000          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   010512                  RESTBL: .WORD   MCERR           ;CONTROLLER ERROR
3001   002166   010623                          .WORD   MDRERR         ;DRIVE ERROR
3002   002170   011150                          .WORD   MNEERR         ;NON-EXISTANT MEMORY ERROR
3003   002172   011122                          .WORD   MFLERR         ;HEADER NOT FOUND-DATA LATE
3004   002174   011105                          .WORD   MHDERR         ;HEADER OR DATA ERROR
3005   002176   011066                          .WORD   MOPERR         ;OPERATION INCOMPLETE
3006   002200   011175                          .WORD   MNDRST         ;NO DRIVE STATUS AVAILABLE
3007   002202   000000                          .WORD   0
3008   002204   011051                          .WORD   MWDERR         ;WRITE DATA ERROR
3009   002206   011033                          .WORD   MHCERR         ;HEAD CURRENT ERROR
3010   002210   000000                          .WORD   0
3011   002212   011015                          .WORD   MSTERR         ;SEEK TIMEOUT ERROR
3012   002214   010762                          .WORD   MSPERR         ;SPINDLE ERROR
3013   002216   011000                          .WORD   MWGERR         ;WRITE GATE ERROR
3014   002220   000000                          .WORD   0
3015   002222   010732                          .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               ;HARDWARE 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/
3356  006015     115  047101  044440  MISTST: .ASCIZ  /MAN INTERVENT STAT/
3357  006040  052123  052101  020105  NSTACHG:        .ASCIZ  /STATE CHG/
3358  006052  050123  042116  020114  SPDERR: .ASCIZ  /SPNDL TIMEOUT FAILED TO SET/
3359  006106  040506  046111  043040  GSTER1: .ASCIZ  /FAIL FORCING DRV SEL ERR/
3360  006137     111  044516  020124  INITST: .ASCIZ  /INIT STATE/
3361  006152  051104  020126  042523  TO5ERR: .ASCIZ  /DRV SELECT/
3362  006165     104  053122  051040  TO9ERR: .ASCIZ  /DRV RDY/
3363  006175     123  042505  020113  T10ERR: .ASCIZ  /SEEK SGN SWITCH/
3364  006215     110  020104  053523  T12ERR: .ASCIZ  /HD SWITCH/
3365  006227     122  020104  042110  T13ERR: .ASCIZ  /RD HDR (P1)/
3366  006243     122  020104  042110  T14ERR: .ASCIZ  /RD HDR (P2)/
3367  006257     127  052122  046040  T16ERR: .ASCIZ  /WRT LCK/
3369  006267                          P2TO1E:
3370  006267     104  043111  020106  P2TO2E: .ASCIZ  /DIFF OF 1 SEEK/
3371  006306  052517  020124  051107  P2TO3E: .ASCIZ  /OUT GRD BAND DETECT/
3372  006332  047111  020103  042523  P2TO4E: .ASCIZ  /INC SEEK FWD HD 0/
3373  006354  047111  020103  042523  P2TO5E: .ASCIZ  /INC SEEK REV HD 0/
3374  006376  047111  020103  042523  P2TO6E: .ASCIZ  /INC SEEK FWD HD 1/
3375  006420  047111  020116  051107  P2TO7E: .ASCIZ  /INN GRD BAND DETECT/
3376  006444  047111  020103  042523  P2TO8E: .ASCIZ  /INC SEEK REV HD 1/
3377  006466  042523  045505     000  P2TO9E: .ASCIZ  /SEEK/
```

```
3378  006473  106     042127  047440  P2T10E: .ASCIZ  /FWD OSC SEEK/
3379  006510  042522  020126  051517  P2T11E: .ASCIZ  /REV OSC SEEK/
3380  006525  123     042505  020113  P2T12E: .ASCIZ  /SEEK TIMING/
3381  006541  102     051501  041511  P2T13E: .ASCIZ  /BASIC READ DATA/
3382  006561  127     052122  051057  P2T14E: .ASCIZ  &WRT/READ DATA (P1)&
3383  006604  050123  047111  046104  P2T15E: .ASCIZ  /SPINDLE ROTATION TIMING/
3384  006634  051127  027524  042522  P2T16E: .ASCIZ  &WRT/READ DATA (P2)&
3385  006657  127     052122  046040  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROTECTION/
3386  006717  101     045104  041440  P2T18E: .ASCIZ  /ADJ CYL INTERFERENCE/
3387  006744  053117  051105  051127  P2T19E: .ASCIZ  /OVERWRITE/
3388  006756  042523  045505  052040  SKTMES: .ASCIZ  /SEEK TIMES /
3389  006772  050123  047111  046104  SRTMES: .ASCIZ  /SPINDLE ROTATION TIME /
3390  007021  050     052123  052101  VALDES: .ASCIZ  /(STATED IN 100'S OF MICRO SEC)/
3391  007060  050101  051120  054117  MAPROX: .ASCIZ  /APPROX /
3392  007070  047111  042516  000122  LABIN:  .ASCIZ  /INNER/
3393  007076  044515  042104  042514  LABMID: .ASCIZ  /MIDDLE/
3394  007105  117     052125  051105  LABOUT: .ASCIZ  /OUTER/
3395  007113  105     050130  041505  LABEXP: .ASCIZ  /EXPECTED/
3396  007124  030060  020061  054503  LABOCF: .ASCIZ  /001 CYL FWD/
3397  007140  030060  020061  054503  LABOCR: .ASCIZ  /001 CYL REV/
3398  007154  031061  020070  054503  LABHCF: .ASCIZ  /128 CYL FWD/
3399  007170  031061  020070  054503  LABHCR: .ASCIZ  /128 CYL REV/
3400  007204  032462  020066  054503  LABACF: .ASCIZ  /256 CYL FWD/
3401  007220  032462  020066  054503  LABACR: .ASCIZ  /256 CYL REV/
3402  007234  042110  020123  040506  HDMOVF: .ASCIZ  /HDS FAILED TO MOVE IN 10 TRIES/
3404  007273  103     046131  050040  CYLPER: .ASCIZ  /CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
3405  007357  110     040505  020104  HAMES1: .ASCIZ  /HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
3406  007442  054524  042520  021040  HAMES2: .ASCIZ  /TYPE "CTL C" & "CONT" TO CONTINUE TESTING/
3407  007514  041101  053117  020105  OPR002: .ASCIZ  /ABOVE CONDITIONS MET/
3408  007541  127     051501  046040  OPR003: .ASCIZ  /WAS LOAD DEPRESSED/
3409  007564  044103  020113  051104  OPR1:   .ASCIZ  /CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
3410  007642  046103  042523  041440  OPR2:   .ASCIZ  /CLSE COVER & RST WRT LCK /
3411  007674  051120  051505  020123  OPR3:   .ASCIZ  /PRESS LOAD /
3412  007710  051120  051505  020123  OPR5:   .ASCIZ  /PRESS LOAD & WAIT FOR LOAD LIGHT /
3413  007752  051120  051505  020123  OPR6:   .ASCIZ  /PRESS LOAD & WAIT FOR RDY /
3414  010005  122     046505  053117  OPR7:   .ASCIZ  /REMOVE ADD PLGS EXCPT /
3415  010034  047111  051123  020124  OPR8:   .ASCIZ  /INSRT ADD PLG /
3416  010053  111     020116  046101  OPR9:   .ASCIZ  /IN ALL DRVS /
3417  010070  047111  052523  043106  OPR10:  .ASCIZ  /INSUFFICIENT DRVS FOR DRV SEL ERR TST/
3418  010136  050122  041514  020105  OPR11:  .ASCIZ  /RPLCE ADD PLGS AS BEFORE/
3420  010167  122     051505  052105  OPR12:  .ASCIZ  /RESET WRT LCK /
3421  010206  047117  000040          OPR1A:  .ASCIZ  /ON /
3422  010212  047117  042040  053122  OPR1B:  .ASCIZ  /ON DRV /
3423  010222  047125  042504  020122  UNDTST: .ASCIZ  /UNDER TEST/
3424  010235  123     052105  053440  OPR004: .ASCIZ  /SET WRT LCK /
3425  010252  044504  043106  000040  DIFWD:  .ASCIZ  /DIFF /
3426  010260  043523  020116  000     SGNWD:  .ASCIZ  /SGN /
3427  010265  110     020104  000     HDWD:   .ASCIZ  /HD /
3428  010271  123     041505  000040  SECWD:  .ASCIZ  /SEC /
3429  010276  054503  020114  000     CYLWD:  .ASCIZ  /CYL /
3430  010303  106     047522  020115  FRMWD:  .ASCIZ  /FROM /
3431  010311  040     054502  040520  BYPSNM: .ASCIZ  / BYPASSED /
3432  010324  047522  052125  047111  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ (IN SEQ CALLED):/
3433  010367  104     053122  051440  STAMES: .ASCIZ  /DRV STAT/
3434  010400  040502  020104  042523  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED GOOD./
3435  010456  047524  020124  047503  TCERR:  .ASCIZ  /TOT COMPARE ERRS: /
```

```
3436
3437                                            ;       RESULT NAMES
3438  010501     104  053122  051040  MDRDY:   .ASCIZ   /DRV RDY /
3439  010512  047503  052116  042440  MCERR:   .ASCIZ   /CONT ERR /
3440  010524  042110  020122  051103  MHCRC:   .ASCIZ   /HDR CRC/
3441  010534  040504  040524  041440  MDCRC:   .ASCIZ   /DATA CRC/
3442  010545     110  051104  047040  MHNF:    .ASCIZ   /HDR NOT FND/
3443  010561     104  052101  020101  MDLT:    .ASCIZ   /DATA LATE/
3444  010573     110  051104  047040  MHFCRC:  .ASCIZ   &HDR NOT FND/HDR CRC/OPI&
3445  010623     104  053122  042440  MDRERR:  .ASCIZ   /DRV ERR /
3447  010634  042523  042514  052103  MHSTA:   .ASCIZ   /SELECTED HD /
3448  010651     126  046117  041440  MVOLCK:  .ASCIZ   /VOL CHK /
3449  010662  047503  042526  020122  MCOSTA:  .ASCIZ   /COVER OPN /
3450  010675     102  052522  044123  MBHSTA:  .ASCIZ   /BRUSH HME /
3451  010710  051127  020124  041514  MWLSTA:  .ASCIZ   /WRT LCK /
3452  010721     110  051504  047440  MHOSTA:  .ASCIZ   /HDS OUT /
3454  010732  051104  020126  042523  MDSERR:  .ASCIZ   /DRV SEL ERR /
3455  010747     104  053122  051440  MDRVST:  .ASCIZ   /DRV STATE /
3456  010762  050123  047111  052040  MSPERR:  .ASCIZ   /SPIN TIMEOUT /
3457  011000  051127  020124  040507  MWGERR:  .ASCIZ   /WRT GAT ERR /
3458  011015     123  042505  020113  MSTERR:  .ASCIZ   /SEEK TIMEOUT /
3459  011033     110  040505  020104  MHCERR:  .ASCIZ   /HEAD CUR ERR /
3460  011051     127  052122  042040  MWDERR:  .ASCIZ   /WRT DAT ERR /
3461  011066  050117  044440  041516  MOPERR:  .ASCIZ   /OP INCOMPLETE /
3462  011105     110  051104  042057  MHDERR:  .ASCIZ   &HDR/DAT ERR &
3463  011122  042110  020122  047516  MFLERR:  .ASCIZ   &HDR NOT FND/DAT LATE &
3464  011150  047516  026516  054105  MNEERR:  .ASCIZ   /NON-EXSTNT MEM /
3465  011170  054503  020114     000  MCYLOC:  .ASCIZ   /CYL /
3466  011175     103  052517  042114  MNDRST:  .ASCIZ   /COULD NOT RETRIEVE DRIVE STATUS/
3467  011235     125  045516  020116  MUNDEF:  .ASCIZ   /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
3468  011302  040506  046111  052040  MRLFAL:  .ASCIZ   /FAIL TO RELD HDS AFTER ERR CLEAR/
3469  011343     127  044522  042524  MWRTAB:  .ASCIZ   /WRITE ABORTED/
3470  011361     040  051105  020122  MEXERS:  .ASCIZ   / ERR LIMIT EXCEEDED - UNIT DROPPED/
3471  011424  042440  051122  051117  MERRS:   .ASCIZ   / ERROR/
3472  011433     207  177777     000  BELL:    .ASCIZ   <207><377><377>
3473
3474                                            ;       RESULT SETTINGS
3475  011437     111  020123     000  RESE3:   .ASCIZ   /IS /
3476  011443     040  041123  000040  RESE4:   .ASCIZ   / SB /
3477
3478                                            ;       RESULT CONDITIONS
3479  011450  044440  020116     000  RESE5:   .ASCIZ   / IN /
3480  011455     040  043117  000040  RESE6:   .ASCIZ   / OF /
3481  011462  052123  052101  020105  STATE2:  .ASCIZ   /STATE 2/
3482  011472  052123  052101  020105  STATE3:  .ASCIZ   /STATE 3/
3483  011502  052123  052101  020105  STATE5:  .ASCIZ   /STATE 5/
3485  011512  042523  045505  053440  CDRDY:   .ASCIZ   &SEEK W/O MOTION&
3487  011532  044506  051522  020124  C10MS:   .ASCIZ   /FIRST 3 MS/
3488  011545     065  030060  051515  C500MS:  .ASCIZ   /500MS/
3489  011553     103  041531  042514  CCYLUP:  .ASCIZ   /CYCLE UP/
3490  011564  040504  040524  054040  CAFDT:   .ASCIZ   /DATA XFER/
3491  011576  020065  042523  042103  C5SEC:   .ASCIZ   /5 SECDS/
3492
3493  011606  047045  052045  047045  FMTOP1:  .ASCIZ   /%N%T%N%T%T%06%S%T%01%N/
3494  011635     045  022516  022524  FMTOP2:  .ASCIZ   /%N%T%01%S1%T%01%N/
3495  011657     045  022516  022524  FMTOP3:  .ASCIZ   /%N%T%01%S1%T%T%N/
```

```
3496  011700  052045  052045     000  FMT1:    .ASCIZ  /%T%T/
3497  011705     045  022516  022524  FMT1.1:  .ASCIZ  /%N%T%T/
3498  011714  052045     000          FMT2:    .ASCIZ  /%T/
3499  011717     045  000116          FMT3:    .ASCIZ  /%N/
3500  011722  047045  052045  052045  FMT4:    .ASCIZ  /%N%T%T%N/
3501  011733     045  022516  022524  FMT5:    .ASCIZ  /%N%T%06%S1%T%01/
3502  011753     045  022516  030523  FMT6:    .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
3503  012015     045  022516  022524  FMT7:    .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
3504  012065     045  022516  022524  FMT8:    .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
3505  012117     045  022516  000124  FMT9:    .ASCIZ  /%N%T/
3506  012124  052045  047445  000061  FMT11:   .ASCIZ  /%T%01/
3507  012132  052045  047445  000063  FMT12:   .ASCIZ  /%T%03/
3508  012140  047045  051445  030461  FMT13:   .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
3509  012204  047045  052045  052045  FMT14:   .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
3510  012236  047045  051445  030461  FMT15:   .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
3511  012272  047045  051445  022465  FMT16:   .ASCIZ  /%N%S5%06/
3512  012303     045  030523  022460  FMT17:   .ASCIZ  /%S10%T%N%S11%06%N/
3513  012325     045  022516  030523  FMT18:   .ASCIZ  /%N%S13%T%S5%T%S4%T%S5%T%N/
3514  012357     045  022524  031123  FMT19:   .ASCIZ  /%T%S2%D6%S4%D6%S4%D6%S4%D6%N/
3515  012414  052045  051445  022462  FMT20:   .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
3516  012444  052045  051445  031061  FMT21:   .ASCIZ  /%T%S12%D6%S14%D6%N/
3517  012467     045  022516  030523  FMT22:   .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
3518  012523     045  022524  022524  FMT23:   .ASCIZ  /%T%T%T%01%N/
3519  012537     045  022516  000124  FMT24:   .ASCIZ  /%N%T/
3520  012544  047045  042045  022462  FMT25:   .ASCIZ  /%N%D2%T/
3521  012554  047045  051445  022461  FMT26:   .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
3522  012600  047045  052045  042045  FMT27:   .ASCIZ  /%N%T%D3%T%D3%N/
3523  012617     045  022516  022524  FMT28:   .ASCIZ  /%N%T%T%T/
3524  012630                          ENDMOD
3529
3530  012630                          BGNMOD   GLBERR
3531                                  ;        ERR1     R3 POINTS TO RESULT MESSAGE
3532                                  ;                 RESULT: (R3)
3533
3534                                  ;        ERR2     R3 POINTS TO RESULT NAME
3535                                  ;                 RESULT: (R3) IS 1 SB 0
3536
3537                                  ;        ERR3     R3 POINTS TO RESULT NAME
3538                                  ;                 RESULT: (R3) IS 0 SB 1
3539
3540                                  ;        ERR4     R3 POINTS TO RESULT NAME
3541                                  ;                 R4 POINTS TO RESULT CONDITIONS
3542                                  ;                 RESULT: (R3) IS 1 SB 0 (R4)
3543
3544                                  ;        ERR5     R3 POINTS TO RESULT NAME
3545                                  ;                 R4 POINTS TO RESULT CONDITIONS
3546                                  ;                 RESULT: (R3) IS 0 SB 1 (R4)
3547
3548                                  ;        ERR6     RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
3549                                  ;                 REPORTS ALL
3550                                  ;                 RESULT: "ERROR" IS 1 SB 0
3551
3552                                  ;        ERR7     DRIVE STATE ERROR REPORT
3553                                  ;                 R3 CONTAINS EXPECTED STATE
3554                                  ;                 T.STAT CONTAINS BAD STATE
3555                                  ;                 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)
                                        ;
012630                                  BGNMSG  ERR1
012630  105737  002657                          TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
012634  001002                                  BNE     1$              ;YES - SKIP
012636  005237  002650                          INC     ERRCNT          ;ELSE BUMP ERROR COUNT
012642  010146                          1$:     MOV     R1,-(SP)        ;STORE R1
012644  004737  021252                          JSR     PC,RPTOP        ;REPORT OPERATION
012650  012721  000001                          MOV     #1,(R1)+        ;SET PARAM NUMBER
012654  010321                                  MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
012656  004737  022040                          JSR     PC,RPTRES       ;REPORT RESULTS
012662  004737  022246                          JSR     PC,RPTREM       ;REPORT REMAINDER
012666  012601                                  MOV     (SP)+,R1        ;RESTORE R1
012670  004737  015656                          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
012674                                  ENDMSG
012674                                  L10000:
012674  104023                                  EMT     C$MSG

012676                                  BGNMSG  ERR2
012676  005237  002650                          INC     ERRCNT          ;BUMP ERROR COUNT
012702  010146                                  MOV     R1,-(SP)        ;STORE R1
012704  004737  021252                          JSR     PC,RPTOP        ;REPORT OPERATION
012710  012721  000003                          MOV     #3,(R1)+        ;SET PARAM NUMBER
012714  010321                                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
012716  012721  000001                          MOV     #1,(R1)+        ;SET IS VALUE
012722  005021                                  CLR     (R1)+           ;SET SB VALUE
012724  004737  022040                          JSR     PC,RPTRES       ;REPORT RESULTS
012730  004737  022246                          JSR     PC,RPTREM       ;REPORT REMAINDER
012734  012601                                  MOV     (SP)+,R1        ;RESTORE R1
012736  004737  015656                          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
012742                                  ENDMSG
012742                                  L10001:
012742  104023                                  EMT     C$MSG

012744                                  BGNMSG  ERR3
012744  005237  002650                          INC     ERRCNT            UMP ERROR COUNT
012750  010146                                  MOV     R1,-(SP)        STORE R1
012752  004737  021252                          JSR     PC,RPTOP        ;REPORT OPERATION
012756  012721  000003                          MOV     #3,(R1)+        ;SET PARAM NUMBER
```

```
3608  012762  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3609  012764  005021                        CLR     (R1)+           ;SET IS VALUE
3610  012766  012721  000001                MOV     #1,(R1)+        ;SET SB VALUE
3611  012772  004737  022040                JSR     PC,RPTRES       ;REPORT RESULTS
3612  012776  004737  022246                JSR     PC,RPTREM       ;REPORT REMAINDER
3613  013002  012601                        MOV     (SP)+,R1        ;RESTORE R1
3614  013004  004737  015656                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3615  013010                        ENDMSG
(3)   013010                        L10002:
(3)   013010  104023                        EMT     C$MSG
3616
3617  013012                        BGNMSG  ERR4
3618  013012  005237  002650                INC     ERRCNT          ;BUMP ERROR COUNT
3619  013016  010146                        MOV     R1,-(SP)        ;STORE R1
3620  013020  004737  021252                JSR     PC,RPTOP        ;REPORT OPERATION
3621  013024  012721  000004                MOV     #4,(R1)+        ;SET PARAM NUMBER
3622  013030  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3623  013032  012721  000001                MOV     #1,(R1)+        ;SET IS VALUE
3624  013036  005021                        CLR     (R1)+           ;SET SB VALUE
3625  013040  010411                        MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
3626  013042  004737  022040                JSR     PC,RPTRES       ;REPORT RESULTS
3627  013046  004737  022246                JSR     PC,RPTREM       ;REPORT REMAINDER
3628  013052  012601                        MOV     (SP)+,R1        ;RESTORE R1
3629  013054  004737  015656                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3630  013060                        ENDMSG
(3)   013060                        L10003:
(3)   013060  104023                        EMT     C$MSG
3631
3632  013062                        BGNMSG  ERR5
3633  013062  005237  002650                INC     ERRCNT          ;BUMP ERROR COUNT
3634  013066  010146                        MOV     R1,-(SP)        ;STORE R1
3635  013070  004737  021252                JSR     PC,RPTOP        ;REPORT OPERATION
3636  013074  012721  000004                MOV     #4,(R1)+        ;SET PARAM NUMBER
3637  013100  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
3638  013102  005021                        CLR     (R1)+           ;SET IS VALUE
3639  013104  012721  000001                MOV     #1,(R1)+        ;SET SB VALUE
3640  013110  010411                        MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
3641  013112  004737  022040                JSR     PC,RPTRES       ;REPORT RESULTS
3642  013116  004737  022246                JSR     PC,RPTREM       ;REPORT REMAINDER
3643  013122  012601                        MOV     (SP)+,R1        ;RESTORE R1
3644  013124  004737  015656                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3645  013130                        ENDMSG
(3)   013130                        L10004:
(3)   013130  104023                        EMT     C$MSG
3646
3647  013132                        BGNMSG  ERR6
3648  013132  105737  002657                TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
3649  013136  001002                        BNE     17$             ;YES - SKIP
3650  013140  005237  002650                INC     ERRCNT          ;ELSE BUMP ERROR COUNT
3651  013144  010146                17$:    MOV     R1,-(SP)        ;STORE R1
3652  013146  010346                        MOV     R3,-(SP)        ;STORE R3
3653  013150  010446                        MOV     R4,-(SP)        ;STORE R4
3654  013152  010546                        MOV     R5,-(SP)        ;STORE R5
3655  013154  004737  021252                JSR     PC,RPTOP        ;REPORT OPERATION
3656  013160  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
3657  013164  012761  000001  000002        MOV     #1,2(R1)        ;INSERT IS VALUE
```

```
 3658  013172  005037  002536                    CLR     TEMP3                   ;CLEAR FOR STATUS STORAGE
 3659  013176  013703  002456                    MOV     T.CS,R3                 ;GET T.CS
 3660  013202  042703  177761                    BIC     #177761,R3              ;AND CLEAR ALL BUT FUNCTION
 3661  013206  022703  000004                    CMP     #4,R3                   ;CHECK IF IT WAS GET STATUS
 3662  013212  001432                            BEQ     1$                      ;YES - STATUS IS IN T.MP, SKIP
 3663  013214  012762  000003  000004            MOV     #GETSTAT,RLDA(R2)       ;ELSE DO GET STATUS
 3664  013222  012703  000004                    MOV     #4,R3
 3665  013226  053703  002444                    BIS     RLDRV,R3
 3666  013232  010362  000000                    MOV     R3,RLCS(R2)
 3667  013236                                    WAITUS  #10.                    ;WAIT FOR CONTROLLER READY
   (3) 013236  012700  000012                    MOV     #10.,R0
   (3) 013242  104027                            EMT     C$WTU
 3668  013244  032762  000200  000000            BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
 3669  013252  001003                            BNE     10$                     ;YES - SKIP
 3670  013254  012703  001000          9$:       MOV     #BIT9,R3                ;ELSE SET NO DRIVE STATUS BIT
 3671  013260  000413                            BR      2$                      ;IN MESSAGE WORD AND SKIP
 3672  013262  016203  000006          10$:      MOV     RLMP(R2),R3             ;STORE STATUS FOR REPORT
 3673  013266  010337  002536                    MOV     R3,TEMP3
 3674  013272  113703  002537                    MOVB    TEMP3+1,R3              ;GET ERROR BITS IN PROPER POSITION
 3675  013276  000402                            BR      13$
 3676  013300  113703  002465          1$:       MOVB    T.MP+1,R3               ;GET ERROR BITS FROM MP REG
 3677  013304  042703  177442          13$:      BIC     #177442,R3              ;CLEAR UNUSED BITS
 3678  013310  013704  002456          2$:       MOV     T.CS,R4                 ;GET ERROR BITS FROM CS REG
 3679  013314  042704  001777                    BIC     #1777,R4                ;CLEAR UNUSED BITS
 3680  013320  050403                            BIS     R4,R3                   ;MAKE ONE WORD OF POSSIBLE ERRORS
 3681  013322  032703  002000                    BIT     #OPIERR,R3              ;TEST IF OPI SET
 3682  013326  001442                            BEQ     115$                    ;NO - SKIP
 3683  013330  032703  010000                    BIT     #HNFERR,R3              ;TEST IF HDR NOT FOUND ERROR
 3684  013334  001026                            BNE     107$                    ;YES - SKIP
 3685  013336  032703  004000                    BIT     #HCRCERR,R3             ;TEST IF HDR CRC ERR
 3686  013342  001020                            BNE     105$                    ;YES - SKIP
 3687  013344  012704  011066                    MOV     #MOPERR,R4              ;SET OPI ALONE MESSAGE
 3688  013350                            100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
  (10) 013350  012746  011424                    MOV     #MERRS,-(SP)
   (9) 013354  010446                            MOV     R4,-(SP)
   (8) 013356  012746  004747                    MOV     #MRSLT,-(SP)
   (7) 013362  012746  012617                    MOV     #FMT28,-(SP)
   (6) 013366  012746  000004                    MOV     #4,-(SP)
   (3) 013372  010600                            MOV     SP,R0
   (4) 013374  104014                            EMT     C$PNTB
   (4) 013376  062706  000012                    ADD     #12,SP
 3689  013402  000430                            BR      120$                    ;SKIP
 3690  013404  012704  010524          105$:     MOV     #MHCRC,R4               ;HDR CRC MESSAGE
 3691  013410  000757                            BR      100$
 3692  013412  032703  004000          107$:     BIT     #HCRCERR,R3             ;TEST IF HCRC WITH HDR NOT FND
 3693  013416  001003                            BNE     109$                    ;YES - SKIP
 3694  013420  012704  010545                    MOV     #MHNF,R4                ;MESSAGE HEADER NOT FOUND
 3695  013424  000751                            BR      100$
 3696  013426  012704  010573          109$:     MOV     #MHFCRC,R4              ;HNF AND HCRC MESSAGE
 3697  013432  000746                            BR      100$                    ;SKIP
 3698  013434  032703  004000          115$:     BIT     #DCKERR,R3              ;TEST IF DATA CHECK SET, NOT OPI
 3699  013440  001403                            BEQ     118$                    ;NO - SKIP
 3700  013442  012704  010534                    MOV     #MDCRC,R4               ;SET MESSAGE DATA CHECK
 3701  013446  000740                            BR      100$                    ;SKIP
 3702  013450  032703  010000          118$:     BIT     #DLTERR,R3              ;TEST IF DATA LATE ERROR
 3703  013454  001403                            BEQ     120$                    ;NO - SKIP
```

```
 3704  013456  012704  010561                    MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
 3705  013462  000732                            BR      100$            ;SKIP
 3706  013464  012705  100000          120$:     MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
 3707  013470  005004                            CLR     R4              ;CLEAR R4 FOR TABLE COUNT
 3708  013472  030503                  3$:       BIT     R5,R3           ;TEST IF BIT IS SET
 3709  013474  001005                            BNE     6$              ;YES - SKIP TO REPORT
 3710  013476  005724                  4$:       TST     (R4)+           ;ELSE BUMP TABLE POINTER
 3711  013500  000241                            CLC                     ;CLEAR CARRY
 3712  013502  006005                            ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
 3713  013504  001372                            BNE     3$              ;LOOP IF NOT 0
 3714  013506  000405                            BR      7$              ;ELSE REPORT REMAINDER
 3715  013510  016411  002164          6$:       MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
 3716  013514  004737  022040                    JSR     PC,RPTRES       ;REPORT RESULTS
 3717  013520  000766                            BR      4$              ;GET NEXT BIT
 3718  013522  004737  022246          7$:       JSR     PC,RPTREM       ;REPORT REMAINDER
 3719  013526  005737  002536                    TST     TEMP3           ;TEST IF ANY NEW STATUS
 3720  013532  001414                            BEQ     15$             ;NO - SKIP
 3721  013534                                    PRINTB  #FMT17,#STAMES,TEMP3
   (9) 013534  013746  002536                    MOV     TEMP3,-(SP)
   (8) 013540  012746  010367                    MOV     #STAMES,-(SP)
   (7) 013544  012746  012303                    MOV     #FMT17,-(SP)
   (6) 013550  012746  000003                    MOV     #3,-(SP)
   (3) 013554  010600                            MOV     SP,R0
   (4) 013556  104014                            EMT     C$PNTB
   (4) 013560  062706  000010                    ADD     #10,SP
 3722  013564  032737  004000  002456  15$:      BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
 3723  013572  001453                            BEQ     25$             ;NO - SKIP
 3724  013574  032737  002000  002456            BIT     #OPIERR,T.CS    ;TEST IF OPI SET
 3725  013602  001047                            BNE     25$             ;YES - SKIP
 3726  013604  005037  002426                    CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
 3727  013610  012701  000200                    MOV     #128.,R1        ;SET COMPARE LENGTH
 3728  013614  012703  000001                    MOV     #1,R3           ;SET WORD COUNT
 3729  013620  012705  003656                    MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
 3730  013624  012704  003256                    MOV     #IBUFF,R4       ;SET TEST WORD POINTER
 3731  013630  021514                  18$:      CMP     (R5),(R4)       ;CHECK WORD
 3732  013632  001427                            BEQ     19$             ;GOOD - SKIP
 3733  013634  023727  002426  000012            CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
 3734  013642  003021                            BGT     20$             ;YES - SKIP
 3735  013644                                    PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
  (13) 013644  011546                            MOV     (R5),-(SP)
  (12) 013646  012746  011443                    MOV     #RESE4,-(SP)
  (11) 013652  011446                            MOV     (R4),-(SP)
  (10) 013654  012746  011437                    MOV     #RESE3,-(SP)
   (9) 013660  010346                            MOV     R3,-(SP)
   (8) 013662  012746  005575                    MOV     #MWORD,-(SP)
   (7) 013666  012746  012236                    MOV     #FMT15,-(SP)
   (6) 013672  012746  000007                    MOV     #7,-(SP)
   (3) 013676  010600                            MOV     SP,R0
   (4) 013700  104014                            EMT     C$PNTB
   (4) 013702  062706  000020                    ADD     #20,SP
 3736  013706  005237  002426          20$:      INC     MORECE          ;BUMP ERROR COUNTER
 3737  013712  022524                  19$:      CMP     (R5)+,(R4)+     ;BUMP POINTERS
 3738  013714  005203                            INC     R3              ;BUMP COUNTER
 3739  013716  005301                            DEC     R1              ;DEC LENGTH COUNT
 3740  013720  001343                            BNE     18$             ;LOOP IF NOT DONE
 3741  013722  005737  002426          25$:      TST     MORECE          ;TEST IF ANY COMPARE ERRORS
```

```
 3742  013726  001421                        BEQ     27$             ;NO - SKIP
 3743  013730  012701  000200                MOV     #128.,R1        ;SET COMPARE LENGTH
 3744  013734                                PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
  (11) 013734  010146                        MOV     R1,-(SP)
  (10) 013736  012746  011455                MOV     #RESE6,-(SP)
  (9)  013742  013746  002426                MOV     MORECE,-(SP)
  (8)  013746  012746  010456                MOV     #TCERR,-(SP)
  (7)  013752  012746  012600                MOV     #FMT27,-(SP)
  (6)  013756  012746  000005                MOV     #5,-(SP)
  (3)  013762  010600                        MOV     SP,R0
  (4)  013764  104014                        EMT     C$PNTB
  (4)  013766  062706  000014                ADD     #14,SP
 3745  013772  012605                27$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
 3746  013774  012604                        MOV     (SP)+,R4
 3747  013776  012603                        MOV     (SP)+,R3
 3748  014000  012601                        MOV     (SP)+,R1
 3749  014002  004737  015656                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 3750  014006                        ENDMSG
  (3)  014006                        L10005:
  (3)  014006  104023                        EMT     C$MSG
 3751
 3752  014010                        BGNMSG  ERR7
 3753  014010  005237  002650                INC     ERRCNT          ;BUMP ERROR COUNT
 3754  014014  010146                        MOV     R1,-(SP)        ;STORE R1
 3755  014016  004737  021252                JSR     PC,RPTOP        ;REPORT OPERATION
 3756  014022  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
 3757  014026  012721  010747                MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
 3758  014032  013721  002472                MOV     T.STAT,(R1)+    ;INSERT IS VALUE
 3759  014036  010311                        MOV     R3,(R1) ;INSERT SB VALUE
 3760  014040  004737  022040                JSR     PC,RPTRES       ;REPORT RESULTS
 3761  014044  004737  022246                JSR     PC,RPTREM       ;REPORT REMAINDER
 3762  014050  012601                        MOV     (SP)+,R1        ;RESTORE R1
 3763  014052  004737  015656                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 3764  014056                        ENDMSG
  (3)  014056                        L10006:
  (3)  014056  104023                        EMT     C$MSG
 3765
 3766  014060                        BGNMSG  ERR8
 3767  014060  005237  002650                INC     ERRCNT          ;BUMP ERROR COUNT
 3768  014064  010146                        MOV     R1,-(SP)        ;STORE R1
 3769  014066  010346                        MOV     R3,-(SP)        ;STORE R3
 3770  014070  004737  021252                JSR     PC,RPTOP        ;REPORT OPERATION
 3771  014074  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
 3772  014100  012721  011170                MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
 3773  014104  013711  002464                MOV     HDWRD1,(R1)     ;GET HEADER WORD
 3774  014110  012703  000007                MOV     #7,R3           ;SET SHIFT COUNT
 3775  014114  000241                3$:     CLC
 3776  014116  006011                        ROR     (R1)            ;ALIGN CHAR FOR PRINTING
 3777  014120  005303                        DEC     R3              ;  AS IS VALUE
 3778  014122  001374                        BNE     3$
 3779  014124  005721                        TST     (R1)+           ;BUMP PARAM POINTER
 3780  014126  013711  002514                MOV     NEWCYL,(R1)     ;INSERT SB VALUE
 3781  014132  004737  022040                JSR     PC,RPTRES       ;REPORT RESULTS
 3782  014136  004737  022246                JSR     PC,RPTREM       ;REPORT REMAINDER
 3783  014142  012603                        MOV     (SP)+,R3        ;RESTORE R3
 3784  014144  012601                        MOV     (SP)+,R1        ;RESTORE R1
```

```
                                            JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                    ENDMSG
                                    L10007:
                                            EMT     C$MSG

                                    BGNMSG  ERR9
                                            INC     ERRCNT          ;BUMP ERROR COUNT
                                            MOV     R1,-(SP)        ;STORE R1
                                            JSR     PC,RPTOP        ;REPORT OPERATION
                                            MOV     #3,(R1)+        ;SET PARAM NUMBER
                                            MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                            MOV     R4,(R1)+        ;SET IS VALUE
                                            MOV     R5,(R1)+        ;SET SB VALUE
                                            JSR     PC,RPTRES       ;REPORT RESULTS
                                            JSR     PC,RPTREM       ;REPORT REMAINDER
                                            MOV     (SP)+,R1        ;RESTORE R1
                                            JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                    ENDMSG
                                    L10010:
                                            EMT     C$MSG

                                    BGNMSG  ERR10
                                            MOV     R1,-(SP)        ;STORE R1
                                            TST     MORECE          ;TEST IF 2ND BAD LINE
                                            BNE     3$              ;YES - SKIP
                                            INC     ERRCNT          ;BUMP ERROR COUNT
                                            JSR     PC,RPTOP        ;REPORT OPERATION
                                            PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
                                            CLR     -(SP)
                                            BISB    RLDRV+1,(SP)
                                            MOV     #DRVNAM,-(SP)
                                            MOV     RLBAS,-(SP)
                                            MOV     #BASADD,-(SP)
                                            MOV     #FMT5,-(SP)
                                            MOV     #5,-(SP)
                                            MOV     SP,R0
                                            EMT     C$PNTB
                                            ADD     #14,SP
                                            PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
                                            MOV     (R5),-(SP)
                                            MOV     #RESE4,-(SP)
                                            MOV     (R4),-(SP)
                                            MOV     #RESE3,-(SP)
                                            MOV     R3,-(SP)
                                            MOV     #MWORD,-(SP)
                                            MOV     #MRSLT,-(SP)
                                            MOV     #FMT14,-(SP)
                                            MOV     #10,-(SP)
                                            MOV     SP,R0
                                            EMT     C$PNTB
                                            ADD     #22,SP
                                            BR      4$
                                    3$:     PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
                                            MOV     (R5),-(SP)
                                            MOV     #RESE4,-(SP)
                                            MOV     (R4),-(SP)
                                            MOV     #RESE3,-(SP)
```

```
    (9)  014366  010346                      MOV     R3,-(SP)
    (8)  014370  012746  005575              MOV     #MWORD,-(SP)
    (7)  014374  012746  012236              MOV     #FMT15,-(SP)
    (6)  014400  012746  000007              MOV     #7,-(SP)
    (3)  014404  010600                      MOV     SP,R0
    (4)  014406  104014                      EMT     C$PNTB
    (4)  014410  062706  000020              ADD     #20,SP
3811     014414  005237  002426      4$:     INC     MORECE          ;INC COMPARE ERROR COUNT
3812     014420  012601                      MOV     (SP)+,R1        ;RESTORE R1
3813     014422  004737  015656              JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
3814     014426                      ENDMSG
  (3)    014426                      L10011:
  (3)    014426  104023                      EMT     C$MSG
3815     014430                      ENDMOD
3816                                         .EVEN
3817
3818     014430                      BGNMOD  HPTCODE
3819     014430                      BGNHW
  (3)    014430  000005                      .WORD   L10012-L$HW/2
3820     014432  174400                      .WORD   174400          ;CSR BASE ADDRESS DEFAULT
3821     014434  000330                      .WORD   330             ;VECTOR DEFAULT
3822     014436  000240                      .WORD   240             ;PRIORITY DEFAULT
3823     014440  000000                      .WORD   0               ;DRIVE NUMBER DEFAULT
3824     014442  000001                      .WORD   1               ;RL11 CONTROLLER
3825     014444                      ENDHW
  (3)    014444                      L10012:
3826     014444                      ENDMOD
3827
3828     014444                      BGNMOD  SPTCODE
3829     014444                      BGNSW
  (3)    014444  000006                      .WORD   L10013-L$SW/2
3830     014446  000000              MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
3831                                                                 ;BIT 1 = USE ALL SECTORS
3832                                                                 ;BIT 2 = EXECUTE DRIVE SELECT TEST
3833                                                                 ;BIT 3 = EXECUTE HEAD ALIGNMENT
3834                                                                 ;BIT 4 = DROP DRIVE IF NO RESPONSE
3835                                                                 ;BIT 12 = HEAD SELECT SUPPLIED FLAG
3836                                                                 ;BIT 13 = HILIMIT SPECIFIED FLAG
3837                                                                 ;BIT 14 = LO LIMIT SPECIFIED FLAG
3838                                                                 ;BIT 15 = DO MANUAL INTERVENTION
3839     014450  000000              LOLIMW: .WORD   0
3840     014452  000377              HILIMW: .WORD   255.
3841     014454  000000              HEADW:  .WORD   0
3842     014456  000024              ERLIMW: .WORD   20.             ;ERROR LIMIT
3843     014460  000012              DCLIMW: .WORD   10.             ;COMPARE ERROR LIMIT
3844     014462                      ENDSW
  (3)    014462                      L10013:
3845     014462                      ENDMOD
3846
3847     014462                      BGNMOD  DSPCODE
3849     014462                      DISPATCH        14
  (4)    014462  000016                      .WORD   14
  (6)    014464  022532                      .WORD   T1
  (6)    014466  023002                      .WORD   T2
  (6)    014470  023204                      .WORD   T3
  (6)    014472  024440                      .WORD   T4
```

```
   (6)  014474  025246                        .WORD   T5
   (6)  014476  025652                        .WORD   T6
   (6)  014500  026524                        .WORD   T7
   (6)  014502  027036                        .WORD   T8
   (6)  014504  027122                        .WORD   T9
   (6)  014506  027422                        .WORD   T10
   (6)  014510  027744                        .WORD   T11
   (6)  014512  030376                        .WORD   T12
   (6)  014514  030714                        .WORD   T13
   (6)  014516  031126                        .WORD   T14
3854    014520                          ENDMOD
3855
3856    014520                          BGNMOD  INITCODE
3857    014520                          BGNINIT
3858    014520                                  SETPRI  #340
   (3)  014520  012700  000340                 MOV     #340,R0
   (3)  014524  104041                         EMT     C$SPRI
3859    014526                                  MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
   (3)  014526  104051                         EMT     C$MANI
3860    014530                                  BCOMPLETE       1$      ;YES - SKIP
   (2)  014530  103403                         BCS     1$
3861    014532  042737  100014  014446         BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
3862                                                                    ; INTERVENTION FLAGS
3863    014540  005037  002414          1$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
3864    014544                                  READEF  #EF.PWR         ;POWER FAILURE
   (3)  014544  012700  000034                 MOV     #EF.PWR,R0
   (3)  014550  104050                         EMT     C$REFG
3865    014552                                  BNCOMPLETE      4$      ;NO, GO CHECK NEW PASS
   (2)  014552  103004                         BCC     4$
3866    014554  013737  002014  002662         MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
3867    014562  000513                         BR      PWCON           ;GO SERVICE POWER FAIL
3868    014564                          4$:     READEF  #EF.START       ;CHECK IF START
   (3)  014564  012700  000040                 MOV     #EF.START,R0
   (3)  014570  104050                         EMT     C$REFG
3869    014572                                  BNCOMPLETE      RESTART ;NO - SKIP
   (2)  014572  103031                         BCC     RESTART
3870                                    ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
3871                                    ;       PASS COUNT, AND ERROR COUNT.
3872    014574  013737  002014  002506         MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
3873    014602  005037  002652          RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
3874    014606  012737  177777  002654         MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
3875    014614  012737  177777  002422         MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
3876    014622  032737  020000  014446         BIT     #HICYL,MISWIW   ;TEST IF HI LIMIT SET
3877    014630  001003                         BNE     3$              ;YES - SKIP
3878    014632  012737  000377  014452         MOV     #377,HILIMW     ;ELSE INIT HILIMIT
3879    014640  032737  040000  014446  3$:    BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
3880    014646  001002                         BNE     5$              ;YES - SKIP
3881    014650  005037  014450                 CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
3882    014654  000427                  5$:    BR      SETDON
3883    014656                          RESTART:
3884    014656                                  READEF  #EF.RESTART     ;CHECK IF RESTART
   (3)  014656  012700  000037                 MOV     #EF.RESTART,R0
   (3)  014662  104050                         EMT     C$REFG
3885    014664                                  BCOMPLETE       RSTRT   ;NO - SKIP
   (2)  014664  103746                         BCS     RSTRT
3886    014666                          CONTINUE:
```

```
 3887  014666                                 READEF  #EF.CONTINUE     ;TEST IF CONTINUE
   (3) 014666  012700  000036                 MOV     #EF.CONTINUE,R0
   (3) 014672  104050                         EMT     C$REFG
 3888  014674                                 BCOMPLETE       PWCON
   (2) 014674  103446                         BCS     PWCON
 3889                               ;         ON CONTINUE PICK UP UNIT LAST UNDER TEST
 3890  014676                                 READEF  #EF.NEW           ;CHECK IF STARTING NEW PASS
   (3) 014676  012700  000035                 MOV     #EF.NEW,R0
   (3) 014702  104050                         EMT     C$REFG
 3891  014704                                 BCOMPLETE       PASNEW
   (2) 014704  103403                         BCS     PASNEW
 3892  014706                           NXTPAS:
 3893  014706  005737  002506                 TST     DRVCNT            ;TEST IF ALL UNITS CHECKED
 3894  014712  001010                         BNE     SETDON            ;NO - SKIP
 3895  014714  005237  002652           PASNEW: INC     PASNUM            ;ELSE BUMP PASS COUNT
 3896  014720  013737  002014  002506         MOV     L$UNIT,DRVCNT     ;GET ALL DRIVES
 3897  014726  012737  177777  002654         MOV     #-1,PSETNM        ;SET PARAM SELECT TO INITIAL
 3898  014734  005237  002654           SETDON: INC     PSETNM            ;NEXT SET OF PARAMETERS
 3899  014740  005337  002506                 DEC     DRVCNT            ;DOWN COUNT DRIVE TOTAL
 3900  014744  013700  002654                 MOV     PSETNM,R0         ;SET UP TO GET PARAMETERS
 3901  014750  012702  002440                 MOV     #RLBAS,R2
 3902  014754                                 GPHARD  R0,R1
   (3) 014754  104042                         EMT     C$GPHRD
   (3) 014756  010001                         MOV     R0,R1
 3903  014760                                 BCOMPLETE       7$        ;SKIP IF GOOD PARAM
   (2) 014760  103406                         BCS     7$
 3904  014762  005737  002662                 TST     PWRFLG            ;RECENT POWER FAILURE
 3905  014766  001747                         BEQ     NXTPAS  ;NO
 3906  014770  005337  002662                 DEC     PWRFLG            ;ACCOUNT FOR DRIVE
 3907  014774  000744                         BR      NXTPAS
 3908  014776  012122                   7$:   MOV     (R1)+,(R2)+       ;STORE PARAMETERS CSR
 3909  015000  012122                         MOV     (R1)+,(R2)+       ;                 VECTOR
 3910  015002  005721                         TST     (R1)+             ;BUMP PAST PRIORITY
 3911  015004  012122                         MOV     (R1)+,(R2)+       ;                 DRIVE
 3912  015006  005037  002650                 CLR     ERRCNT            ;CLEAR OUT ERROR COUNT
 3913  015012                           PWCON:  SETVEC  RLVEC,#INTHLR,#340        ;SET UP VECTOR
   (7) 015012  012746  000340                 MOV     #340,-(SP)
   (6) 015016  012746  015620                 MOV     #INTHLR,-(SP)
   (5) 015022  013746  002442                 MOV     RLVEC,-(SP)
   (4) 015026  012746  000003                 MOV     #3,-(SP)
   (3) 015032  104037                         EMT     C$$VEC
   (2) 015034  062706  000010                 ADD     #10,SP
 3914  015040                                 SETPRI  #0                ;SET PRIORITY
   (3) 015040  012700  000000                 MOV     #0,R0
   (3) 015044  104041                         EMT     C$$PRI
 3915  015046  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  015052  005737  002652                 TST     PASNUM            ;TEST IF PASS 0
 3922  015056  001135                         BNE     22$               ;NO - SKIP
 3923  015060  032737  000020  014446         BIT     #AUTOSZ,MISWIW    ;TEST IF DOING AUTO SIZE
 3924  015066  001531                         BEQ     22$               ;NO - SKIP
 3925                           ;CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
```

```
                                                CLR     TRPFLG                  ;TRAP OCCURANCE
                                                SETVEC  ERRVEC,#TRPHAN,#340     ;SET TRAP VECTOR
  (7)  015074  012746  000340                   MOV     #340,-(SP)
  (6)  015100  012746  015612                   MOV     #TRPHAN,-(SP)
  (5)  015104  013746  002642                   MOV     ERRVEC,-(SP)
  (4)  015110  012746  000003                   MOV     #3,-(SP)
  (3)  015114  104037                           EMT     C$SVEC
  (2)  015116  062706  000010                   ADD     #10,SP
3928   015122  005762  000000                   TST     RLCS(R2)                ;ACCESS BUS
3929   015126  005737  002660                   TST     TRPFLG                  ;TRAP OCCUR??
3930   015132  001032                           BNE     5$                      ;YES, DON'T INVESTIGATE FURTHER
3931   015134  013705  002444                   MOV     RLDRV,R5                ;GET DRIVE NUMBER
3932   015140  052705  000200                   BIS     #CRDYMSK,R5             ;INSERT CONT READY
3933   015144  010562  000000                   MOV     R5,RLCS(R2)             ;LOAD IN DRIVE NUMBER
3934   015150  032762  000001  000000           BIT     #DRDYMSK,RLCS(R2)       ;CHECK IF DRIVE IS READY
3935   015156  001072                           BNE     20$                     ;YES - GO DO TEST
3936   015160  012762  000003  000004           MOV     #GETSTAT,RLDA(R2)       ;ELSE INSERT GET STATUS
3937   015166  052705  000004                   BIS     #4,R5                   ;LOAD R5 WITH GET STATUS FUNCTION
3938   015172  042705  000200                   BIC     #CRDYMSK,R5             ;CLEAR CONTROLLER READY
3939   015176  010562  000000                   MOV     R5,RLCS(R2)             ;LOAD CS REG
3940   015202                                   WAITMS  #4                      ;WAIT 4 MS
  (3)  015202  012700  000004                   MOV     #4,R0
  (3)  015206  104026                           EMT     C$WTM
3941   015210  032762  002000  000000           BIT     #OPIERR,RLCS(R2)        ;TEST IF OPI SET
3942   015216  001452                           BEQ     20$                     ;NO - SKIP
3943   015220                           5$:     CLRVEC  ERRVEC
  (3)  015220  013700  002642                   MOV     ERRVEC,R0
  (3)  015224  104036                           EMT     C$CVEC
3944   015226                                   PRINTF  #FMT24,#DRVNAV
  (8)  015226  012746  005430                   MOV     #DRVNAV,-(SP)
  (7)  015232  012746  012537                   MOV     #FMT24,-(SP)
  (6)  015236  012746  000002                   MOV     #2,-(SP)
  (3)  015242  010600                           MOV     SP,R0
  (4)  015244  104017                           EMT     C$PNTF
  (4)  015246  062706  000006                   ADD     #6,SP
3945   015252                           10$:    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  015252  005046                           CLR     -(SP)
 (11)  015254  153716  002445                   BISB    RLDRV+1,(SP)
 (10)  015260  012746  005423                   MOV     #DRVNAM,-(SP)
  (9)  015264  013746  002440                   MOV     RLBAS,-(SP)
  (8)  015270  012746  005412                   MOV     #BASADD,-(SP)
  (7)  015274  012746  011733                   MOV     #FMT5,-(SP)
  (6)  015300  012746  000005                   MOV     #5,-(SP)
  (3)  015304  010600                           MOV     SP,R0
  (4)  015306  104017                           EMT     C$PNTF
  (4)  015310  062706  000014                   ADD     #14,SP
3946   015314                                   PRINTF  #FMT3
  (7)  015314  012746  011717                   MOV     #FMT3,-(SP)
  (6)  015320  012746  000001                   MOV     #1,-(SP)
  (3)  015324  010600                           MOV     SP,R0
  (4)  015326  104017                           EMT     C$PNTF
  (4)  015330  062706  000004                   ADD     #4,SP
3947   015334                                   DODU    PSETNM                  ;DROP DRIVE
  (3)  015334  013700  002654                   MOV     PSETNM,R0
  (3)  015340  104053                           EMT     C$DODU
3948   015342                                   DOCLN
```

```
  (3)  015342  104044                          EMT     C$DCLN
3949   015344                        20$:      CLRVEC  ERRVEC
  (3)  015344  013700  002642                  MOV     ERRVEC,R0
  (3)  015350  104036                          EMT     C$CVEC
3950   015352                        22$:
3952
3953   015352                                  MANUAL                          ;MANUAL INTERVENTION ALLOWED
  (3)  015352  104051                          EMT     C$MANI
3954   015354                                  BNCOMPLETE      4$              ;NO
  (2)  015354  103004                          BCC     4$
3955
3956   015356  005737  002652                  TST     PASNUM                  ;YES, CHECK PASS NUMBER
3957   015362  001001                          BNE     4$                      ;NOT FIRST PASS, NEED DRIVE UP
3958   015364  000437                          BR      8$                      ;FIRST PASS, PROGRAM WILL INSTRUCT USER
3959
3961                                  ;CHECK IF POWER FAILURE WAIT IS NEEDED
3962
3963   015366  005737  002662          4$:     TST     PWRFLG                  ;NEEDED???
3964   015372  001434                          BEQ     8$                      ;NO, SKIP
3965
3966   015374  013705  002444                  MOV     RLDRV,R5                ;DRIVE SELECT
3967   015400  052705  000200                  BIS     #CRDYMSK,R5             ;SET CRDY
3968   015404  010562  000000                  MOV     R5,RLCS(R2)             ;SELECT DRIVE
3969   015410  012701  000074                  MOV     #60.,R1                 ;SIXTY SECOND TIMER
3970   015414  032762  000001  000000 .9$:     BIT     #DRDYMSK,RLCS(R2)               ;DRIVE UP YET
3971   015422  001020                          BNE     8$                      ;YES START TEST
3972
3973   015424                                  WAITMS  #10.                    ;WAIT A SECOND
  (3)  015424  012700  000012                  MOV     #10.,R0
  (3)  015430  104026                          EMT     C$WTM
3974   015432  005301                          DEC     R1                      ;SIXTY GONE BY
3975   015434  001367                          BNE     9$                      ;NO
3976   015436                                  PRINTF  #FMT24,#NOPWR
  (8)  015436  012746  005463                  MOV     #NOPWR,-(SP)
  (7)  015442  012746  012537                  MOV     #FMT24,-(SP)
  (6)  015446  012746  000002                  MOV     #2,-(SP)
  (3)  015452  010600                          MOV     SP,R0
  (4)  015454  104017                          EMT     C$PNTF
  (4)  015456  062706  000006                  ADD     #6,SP
3977   015462  000673                          BR      10$
3978
3979   015464                        8$:
3980
3981   015464                        ENDINIT
  (3)  015464                        L10014:
  (3)  015464  104011                          EMT     C$INIT
3982   015466                        ENDMOD
3983
3984   015466                        BGNMOD  CLNCODE
3985   015466                        BGNCLN
3986
3987   015466                                  SETVEC  ERRVEC,#TRPHAN,#340
  (7)  015466  012746  000340                  MOV     #340,-(SP)
  (6)  015472  012746  015612                  MOV     #TRPHAN,-(SP)
  (5)  015476  013746  002642                  MOV     ERRVEC,-(SP)
  (4)  015502  012746  000003                  MOV     #3,-(SP)
```

```
  (3) 015506 104037                          EMT     C$SVEC
  (2) 015510 062706 000010                   ADD     #10,SP
3988
3989 015514                                  SETPRI #7                ;SET PRORITY TO 7
  (3) 015514 012700 000007                   MOV     #7,R0
  (3) 015520 104041                          EMT     C$SPRI
3990 015522 032762 000200 000000 2$:         BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
3991 015530 001407                           BEQ     3$                      ;NO LOOP UNTIL READY
3992 015532 053762 002444 000000             BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
3993 015540 032762 000001 000000             BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
3994 015546 001003                           BNE     5$                      ;NO - SKIP
3995 015550                                  3$:     WAITMS  #3              ;WAIT 300 MS
  (3) 015550 012700 000003                   MOV     #3,R0
  (3) 015554 104026                          EMT     C$WTM
3996 015556                                  5$:     CLRVEC  RLVEC           ;RELEASE VEC
  (3) 015556 013700 002442                   MOV     RLVEC,R0
  (3) 015562 104036                          EMT     C$CVEC
3997 015564 005737 002662                    TST     PWRFLG  ;PWR FAIL SET
3998 015570 001402                           BEQ     7$                      ;NO
3999 015572 005337 002662                    DEC     PWRFLG
4000 015576                                  7$:     CLRVEC  ERRVEC
  (3) 015576 013700 002642                   MOV     ERRVEC,R0
  (3) 015602 104036                          EMT     C$CVEC
4001 015604                                  ENDCLN
  (3) 015604                                 L10015:
  (3) 015604 104012                          EMT     C$CLEAN
4002
4003 015606                                  BGNDU
4004 015606 000240                           NOP
4005 015610                                  ENDDU
  (3) 015610                                 L10016:
  (3) 015610 104055                          EMT     C$DU
4006
4007 015612                                  ENDMOD
4008 015612                                  BGNMOD  GLBSUB
4009
4010 015612 005237 002660                    TRPHAN: INC     TRPFLG
4011 015616 000002                           RTI
4012
4013 015620                                  BGNSRV  INTHLR
4014                                         ;       INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
4015 015620                                          ABORTWAIT
  (3) 015620 104021                          EMT     C$ABRT
4016 015622 012237 002456                    MOV     (R2)+,T.CS      ;STORE RL REGISTERS
4017 015626 012237 002460                    MOV     (R2)+,T.BA
4018 015632 012237 002462                    MOV     (R2)+,T.DA
4019 015636 011237 002464                    MOV     (R2),T.MP
4020 015642 012737 177777 002420             MOV     #-1,DONE        ;SET DONE FLAG
4021 015650 013702 002440                    MOV     RLBAS,R2        ;RESTORE R2
4022 015654                                  ENDSRV
  (3) 015654                                 L10017:
  (2) 015654 000002                          RTI
4023
4024                                         ;       ERROR LIMIT CHECKING ROUTINE
4025                                         ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
4026 015656 023737 002650 014456 CKERLM: CMP     ERRCNT,ERLIMW   ;TEST IF ERROR LIMIT EXCEEDED
```

```
 4027  015664  002453                          BLT     1$                      ;NO - SKIP
 4028  015666 -                                INLOOP                          ;CHECK IF IN ERROR LOOP
   (3) 015666  104020                          EMT     C$INLP
 4029  015670                                  BCOMPLETE       1$              ;YES - SKIP
   (2) 015670  103451                          BCS     1$
 4030  015672                                  PRINTF  #FMT25,ERLIMW,#MEXERS
   (9) 015672  012746  011361                  MOV     #MEXERS,-(SP)
   (8) 015676  013746  014456                  MOV     ERLIMW,-(SP)
   (7) 015702  012746  012544                  MOV     #FMT25,-(SP)
   (6) 015706  012746  000003                  MOV     #3,-(SP)
   (3) 015712  010600                          MOV     SP,R0
   (4) 015714  104017                          EMT     C$PNTF
   (4) 015716  062706  000010                  ADD     #10,SP
 4031  015722                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11) 015722  005046                          CLR     -(SP)
  (11) 015724  153716  002445                  BISB    RLDRV+1,(SP)
  (10) 015730  012746  005423                  MOV     #DRVNAM,-(SP)
   (9) 015734  013746  002440                  MOV     RLBAS,-(SP)
   (8) 015740  012746  005412                  MOV     #BASADD,-(SP)
   (7) 015744  012746  011733                  MOV     #FMT5,-(SP)
   (6) 015750  012746  000005                  MOV     #5,-(SP)
   (3) 015754  010600                          MOV     SP,R0
   (4) 015756  104017                          EMT     C$PNTF
   (4) 015760  062706  000014                  ADD     #14,SP
 4032  015764                                  PRINTF  #FMT3
   (7) 015764  012746  011717                  MOV     #FMT3,-(SP)
   (6) 015770  012746  000001                  MOV     #1,-(SP)
   (3) 015774  010600                          MOV     SP,R0
   (4) 015776  104017                          EMT     C$PNTF
   (4) 016000  062706  000004                  ADD     #4,SP
 4033  016004                                  DODU    PSETNM                  ;DROP DRIVE
   (3) 016004  013700  002654                  MOV     PSETNM,R0
   (3) 016010  104053                          EMT     C$DODU
 4034  016012                                  DOCLN                           ;GO TO CLEAN UP
   (3) 016012  104044                          EMT     C$DCLN
 4035  016014  000207              1$:         RTS     PC
 4036
 4037                                          ;
                                               READ AND STORE ALL RL11 REGISTERS
 4038  016016  016237  000000  002456  READRL: MOV     RLCSR(R2),T.CS          ;GET CS REG
 4039  016024  016237  000002  002460          MOV     RLBA(R2),T.BA           ;GET BUS ADDRESS REG
 4040  016032  016237  000004  002462          MOV     RLDA(R2),T.DA           ;GET DISK ADDRESS
 4041  016040  016237  000006  002464          MOV     RLMP(R2),T.MP           ;GET MULTI-PURPOSE REG
 4042  016046  000207                          RTS     PC                      ;RETURN
 4043
 4044                                          ;
                                               WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
 4045  016050  011646              WAITIN: MOV     (SP),-(SP)              ;MAKE ROOM FOR ERROR POINTER
 4046  016052  005066  000002                  CLR     2(SP)                   ;CLEAR FOR POINTER
 4047  016056  032762  000200  000000          BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
 4048  016064  001420                          BEQ     4$                      ;NO - SKIP TO WAIT
 4049  016066  004737  016016                  JSR     PC,READRL               ;READ ALL RL REGS
 4050  016072  005737  002420                  TST     DONE                    ;TEST IF INTERRUPT OCCURRED
 4051  016076  001433                          BEQ     5$                      ;NO - GO SET NO INTERRUPT ERR FLAG
 4052  016100  012766  005603  000002  1$:     MOV     #MTOSLOW,2(SP)          ;ELSE SET TO SLOW ERROR POINTER
 4053  016106  032737  002000  002456          BIT     #OPIERR,T.CS            ;TEST IF OPI SET
 4054  016114  001403                          BEQ     2$                      ;NO - SKIP
 4055  016116  012766  005622  000002          MOV     #MDRRES,2(SP)           ;SET MESSAGE FOR NO DRIVE RESPONSE
```

```
                                                2$:     RTS     PC              ;RETURN
                                                4$:     WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
                                                        MOV     #3,R0
                                                        EMT     C$WTM
                                                        BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
                                                        BNE     3$              ;YES - SKIP
                                                        JSR     PC,READRL       ;READ RL REGS
                                                        MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
                                                        BR      2$              ;SKIP
                                                3$:     TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
                                                        BNE     1$              ;YES - SKIP TO SET TO SLOW
                                                5$:     JSR     PC,READRL       ;READ RL REGS
                                                        MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
                                                        BR      2$              ;GO TO RETURN

                                                ;
                                                ; OPERATION AND TEST INITIALIZE ROUTINE
                                                ;
                                                TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
                                                        CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
                                                        CLR     MORECE          ;RESET MORE COMPARE ERRORS
                                                        RTS     PC

                                                ;
                                                ; GET STATUS AND GET STATUS WITH RESET ROUTINE
                                                ;
                                                GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
                                                        MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
                                                        BR      GSTATG
                                                GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
                                                        MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
                                                        BR      GSTATG
                                                GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
                                                        CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
                                                GSTATG: MOV     R3,-(SP)        ;STORE R3
                                                        MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
                                                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
                                                        MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
                                                        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
                                                        MOV     R3,SSINDX       ;STORE IT BACK
                                                        MOV     R0,-(SP)        ;STORE R0
                                                        MOV     R1,-(SP)        ;STORE R1
                                                        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
                                                        BIT     #DRSET,TEMP4    ;TEST IF DRIVE RESET
                                                        BEQ     11$             ;NO - SKIP
                                                        BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
                                                        BEQ     49$             ;NO - SKIP
                                                        WAITMS  #3              ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
                                                        MOV     #3,R0
                                                        EMT     C$WTM
                                                49$:    MOV     #50.,R1         ;SET WAIT FOR 5 SEC
                                                50$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
                                                        3$
                                                        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
                                                        BNE     5$              ;YES - GO DO CLEAR
                                                        BIT     #HOSTAT,T.MP    ;ELSE TEST IF HEADS OUT
                                                        BNE     51$             ;YES - BYPASS RELOAD WAIT FLAG SETTING
                                                        BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
                                                                                        ;THAT CAUSED HEADS TO
                                                                                        ;UNLOAD
```

```
 4108  016410  001441                        BEQ     5$              ;NO - SKIP
 4109  016412  052737  040000  002416        BIS     #RELDWT,OPFLAG  ;ELSE SET WAIT FLAG
 4110  016420  000435                        BR      5$              ;SKIP TO CLEAR
 4111  016422  032737  040000  002456  51$:  BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR NOW
 4112  016430  001031                        BNE     5$              ;YES - SKIP TO CLEAR
 4113  016432                                WAITMS  #1              ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HO
   (3) 016432  012700  000001                MOV     #1,R0
   (3) 016436  104026                        EMT     C$WTM
 4114  016440  005301                        DEC     R1              ;DEC WAIT COUNTER
 4115  016442  001344                        BNE     50$             ;IF NOT DONE, LOOP
 4116  016444  012703  011235                MOV     #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
 4117  016450                                ERRHRD  10001.,,ERR1
   (3) 016450  104443                        TRAP    T$ERCODE
   (5) 016452  023421                        .WORD   10001
   (5) 016454  012630                        .WORD   ERR1
 4118  016456  000554                        BR      14$             ;EXIT
 4119  016460  005737  002540          11$:  TST     TEMP4           ;TEST IF SAVE REGISTERS
 4120  016464  001013                        BNE     5$              ;NO SKIP
 4121  016466  012701  000004                MOV     #4,R1           ;SET SAVE COUNT
 4122  016472  012703  002456                MOV     #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
 4123  016476  014346                  8$:   MOV     -(R3),-(SP)     ;PUT REG ON STACK
 4124  016500  005301                        DEC     R1              ;DEC COUNT
 4125  016502  001375                        BNE     8$              ;LOOP UNTIL ALL SAVED
 4126  016504  012737  000003  002452        MOV     #GETSTAT,L.DA   ;SET FOR GET STATUS
 4127  016512  000403                        BR      6$              ;SKIP
 4128  016514  013737  002540  002452  5$:   MOV     TEMP4,L.DA      ;INSERT PRESET FOR STATUS
 4129  016522                          6$:
 4130  016522  005037  002420                CLR     DONE            ;CLEAR INTERRUPT FLAG
 4131  016526  013737  002444  002446        MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
 4132  016534  042737  002000  002446        BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4133  016542  052737  000104  002446        BIS     #GTSTAT,L.CS
 4134  016550  013762  002452  000004        MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
 4135  016556  013762  002446  000000        MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
 4136  016564                                WAITUS  #1              ;WAIT 100 US FOR INTERRUPT
   (3) 016564  012700  000001                MOV     #1,R0
   (3) 016570  104027                        EMT     C$WTU
 4137  016572  005737  002420                TST     DONE            ;CHECK IF INTERRUPT OCCURRED
 4138  016576  001476                        BEQ     1$              ;NO - SKIP
 4139  016600  013737  002464  002472  4$:   MOV     T.MP,T.STAT     ;STORE MP REGISTER
 4140  016606  042737  177770  002472        BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
 4141  016614  032737  000010  002452        BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
 4142  016622  001474                        BEQ     3$              ;NO - SKIP TO EXIT
 4143  016624  032737  040000  002416        BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
 4144  016632  001424                        BEQ     12$             ;NO - SKIP
 4145  016634  012701  001130                MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
 4146  016640  032762  000001  000000  13$:  BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
 4147  016646  001016                        BNE     12$             ;YES - SKIP
 4148  016650                                WAITMS  #1              ;CALL WAIT
   (3) 016650  012700  000001                MOV     #1,R0
   (3) 016654  104026                        EMT     C$WTM
 4149  016656  005301                        DEC     R1              ;DEC COUNT
 4150  016660  001367                        BNE     13$             ;LOOP IF NOT 0
 4151  016662  004737  016250                JSR     PC,GSTAT        ;GET DRIVE STATUS
 4152  016666  017014                        3$                      ;ERROR RETURN
 4153  016670  012703  011302                MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
 4154  016674                                ERRHRD  10003.,,ERR1
```

```
  (3)  016674  104443                          TRAP    T$ERCODE
  (5)  016676  023423                          .WORD   10003
  (5)  016700  012630                          .WORD   ERR1
 4155  016702  000442                          BR      14$             ;GO TO EXIT
 4156  016704                          12$:    WAITUS  #10.            ;WAIT FOR 1MS
  (3)  016704  012700  000012                  MOV     #10.,R0
  (3)  016710  104027                          EMT     C$WTU
 4157  016712  004737  016250                  JSR     PC,GSTAT        ;GET DRIVE STATUS
 4158  016716  017014                          3$
 4159  016720  032737  100000  002456          BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
 4160  016726  001432                          BEQ     3$              ;NO - SKIP
 4161  016730  032737  001000  002464          BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
 4162  016736  001403                          BEQ     7$              ;YES SKIP
 4163  016740  012703  005751                  MOV     #VCNRST,R3      ;SET REASON POINTER
 4164  016744  000416                          BR      2$              ;EXIT
 4165  016746  032737  040000  002456  7$:     BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
 4166  016754  001404                          BEQ     9$              ;NO - SKIP
 4167  016756                                  ERRHRD  10004.,ERR6
  (3)  016756  104443                          TRAP    T$ERCODE
  (5)  016760  023424                          .WORD   10004
  (5)  016762  013132                          .WORD   ERR6
 4168  016764  000411                          BR      14$             ;EXIT
 4169  016766  012703  005772          9$:     MOV     #UNXERR,R3      ;SET REASON POINTER
 4170  016772  000403                          BR      2$              ;EXIT
 4171  016774  004737  016050          1$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 4172  017000  012603                          MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
 4173  017002                          2$:     ERRHRD  10002.,ERR1
  (3)  017002  104443                          TRAP    T$ERCODE
  (5)  017004  023422                          .WORD   10002
  (5)  017006  012630                          .WORD   ERR1
 4174  017010  005037  002430          14$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4175  017014  005737  002540          3$:     TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 4176  017020  001007                          BNE     22$             ;NO - SKIP
 4177  017022  012703  002446                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE
 4178  017026  012701  000004                  MOV     #4,R1           ;SET REGISTER COUNT
 4179  017032  012623                  20$:    MOV     (SP)+,(R3)+     ;RESTORE REG
 4180  017034  005301                          DEC     R1              ;DEC COUNT
 4181  017036  001375                          BNE     20$             ;LOJP UNTIL ALL ARE RESTORED
 4182  017040  162737  000002  002414  22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4183  017046  012601                          MOV     (SP)+,R1        ;RESTORE R1
 4184  017050  012600                          MOV     (SP)+,R0
 4185  017052  012603                          MOV     (SP)+,R3        ;RESTORE R3
 4186  017054  012637  002540                  MOV     (SP)+,TEMP4     ;RESTORE TEMP4
 4187  017060  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4188  017064  001403                          BEQ     99$             ;YES - SKIP
 4189  017066  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4190  017072  000207                          RTS     PC
 4191  017074  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4192  017100  000207                          RTS     PC
 4193
 4194
 4293  017102  010346                  SIMSEK: MOV     R3,-(SP)        ;STORE REGISTERS
 4294  017104  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4295  017110  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4296  017112  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4297  017120  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
```

```
4298  017126  010337  002414                    MOV     R3,SSINDX       ;STORE IT BACK
4299  017132  010046                            MOV     R0,-(SP)
4300  017134  010446                            MOV     R4,-(SP)
4301  017136  012737  000002  002430            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
4302  017144  004737  017366                    JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
4303  017150  017330                            #65$
4304  017152  012704  002446                    MOV     #L.CS,R4        ;GET POINTER TO L REGS
4305  017156  012714  000106                    MOV     #SEEK,(R4)      ;SET FOR SEEK
4306  017162  053714  002444                    BIS     RLDRV,(R4)      ;INSERT DRIVE NUMBER
4307  017166  042724  002000                    BIC     #BIT10,(R4)+    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
4308  017172  005024                            CLR     (R4)+           ;CLEAR BUS ADDRESS
4309  017174  013714  002520                    MOV     DESDIF,(R4)     ;LOAD DIFFERENCE
4310  017200  012703  000007                    MOV     #7,R3           ;SET COUNT FOR SHIFT TO ALIGN
4311  017204  006314                    3$:     ASL     (R4)            ;ALIGN DIFFERENCE IN DA
4312  017206  005303                            DEC     R3
4313  017210  001375                            BNE     3$
4314  017212  005737  002522                    TST     DESSGN          ;TEST IF SIGN SET
4315  017216  001402                            BEQ     5$              ;NO - SKIP
4316  017220  052714  000004                    BIS     #DIRBIT,(R4)    ;INSERT SIGN
4317  017224  005737  002524            5$:     TST     DESHD           ;TEST IF HEAD 0
4318  017230  001402                            BEQ     7$              ;YES - SKIP
4319  017232  052714  000020                    BIS     #HDSEL,(R4)     ;INSERT HEAD BIT
4320  017236  052724  000001            7$:     BIS     #MBSET0,(R4)+   ;INSERT MARKER BIT
4321  017242  005037  002420                    CLR     DONE            ;CLEAR INTERRUPT FLAG
4322  017246  012701  000012                    MOV     #10.,R1         ;SET WAIT COUNT FOR 800US
4323  017252  014462  000004                    MOV     -(R4),RLDA(R2)  ;LOAD RL REGISTERS
4324  017256  014462  000002                    MOV     -(R4),RLBA(R2)
4325  017262  014462  000000                    MOV     -(R4),RLCS(R2)
4326  017266  005737  002420            10$:    TST     DONE            ;CKECK IF INTERRUPTED
4327  017272  001016                            BNE     65$             ;YES - SKIP
4328  017274  005301                            DEC     R1              ;DEC WAIT COUNT
4329  017276  001404                            BEQ     13$             ;IF 0 - SKIP
4330  017300                                    WAITUS  #1
  (3) 017300  012700  000001                    MOV     #1,R0
  (3) 017304  104027                            EMT     C$WTU
4331  017306  000767                            BR      10$             ;GO CHECK DONE
4332  017310  004737  016050            13$:    JSR     PC,WAITIN       ;GO WAIT FOR TIMEOUT
4333  017314  012603                            MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
4334  017316                                    ERRHRD  10011,,ERR1
  (3) 017316  104443                            TRAP    T$ERCODE
  (5) 017320  023433                            .WORD   10011
  (5) 017322  012630                            .WORD   ERR1
4335  017324  005037  002430                    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
4336  017330                            14$:
4337  017330  162737  000002  002414  65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
4338  017336  012604                            MOV     (SP)+,R4        ;RESTORE REGS
4339  017340  012600                            MOV     (SP)+,R0
4340  017342  012603                            MOV     (SP)+,R3
4341  017344  005737  002430                    TST     ERRSWI          ;TEST IF ERROR RETURN
4342  017350  001403                            BEQ     99$             ;YES - SKIP
4343  017352  063716  002430                    ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
4344  017356  000207                            RTS     PC
4345  017360  017616  000000            99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
4346  017364  000207                            RTS     PC
4348
4424                                      ;     DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT
```

```
                                        ;
                                        ;500MS FOR READY TO SET.
                                RDYCHK: MOV     R3,-(SP)        ;STORE REGS
                                        MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
                                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
                                        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
                                        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
                                        MOV     R3,SSINDX       ;STORE IT BACK
                                        MOV     R0,-(SP)
                                        MOV     R1,-(SP)
                                        MOV     R4,-(SP)
                                        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
                                        MOV     #5000.,R1       ;SET WAIT COUNT
                                1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
                                        4$
                                        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
                                        BNE     5$              ;YES - EXIT
                                        WAITUS  #1
                                        MOV     #1,R0
                                        EMT     C$WTU
                                        DEC     R1              ;DEC WAIT COUNT
                                        BNE     1$              ;LOOP IF NOT 0
                                        MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
                                        MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
                                        ERRHRD  10010.,,ERR5
                                        TRAP    T$ERCODE
                                        .WORD   10010
                                        .WORD   ERR5
                                        MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
                                2$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
                                        4$
                                        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
                                        BNE     3$              ;YES - SKIP
                                        WAITMS  #1              ;WAIT FOR 100MS
                                        MOV     #1,R0
                                        EMT     C$WTM
                                        DEC     R1              ;DEC WAIT COUNTER
                                        BNE     2$              ;LOOP UNTIL TIME DONE
                                3$:     BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
                                        BEQ     4$              ;NO - SKIP
                                        ERRHRD  10011.,,ERR6    ;REPORT ALL ERRORS
                                        TRAP    T$ERCODE
                                        .WORD   10011
                                        .WORD   ERR6
                                        DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
                                4$:     CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
                                5$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
                                        MOV     (SP)+,R4        ;RESTORE REGS
                                        MOV     (SP)+,R1
                                        MOV     (SP)+,R0
                                        MOV     (SP)+,R3
                                        TST     ERRSWI          ;TEST IF ERROR RETURN
                                        BEQ     99$             ;YES - SKIP
                                        ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                        RTS     PC
                                99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                        RTS     PC
```

```
 4471
 4514                                           ;
                                                READ HEADERS ROUTINE.
 4515  017626  012737  000001  002540  XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
 4516  017634  000402                          BR      XRDHDG          ;GO DO IT
 4517  017636  005037  002540          XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
 4518  017642  010346                  XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
 4519  017644  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4520  017650  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4521  017652  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4522  017660  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4523  017666  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4524  017672  010046                          MOV     R0,-(SP)
 4525  017674  010146                          MOV     R1,-(SP)
 4526  017676  010446                          MOV     R4,-(SP)
 4527  017700  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4528  017706  005737  002540                  TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
 4529  017712  001007                          BNE     2$              ;NO - SKIP
 4530  017714  012703  002456                  MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
 4531  017720  012701  000004                  MOV     #4,R1           ;SET COUNT
 4532  017724  014346                  1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
 4533  017726  005301                          DEC     R1              ;DEC COUNT
 4534  017730  001375                          BNE     1$              ;LOOP UNTIL ALL ARE SAVED
 4535  017732  004737  017366          2$:     JSR     PC,RDYCHK       ;CHECK DRIVE READY
 4536  017736  020172                          #65$
 4537  017740  005037  002420                  CLR     DONE            ;CLEAR INTERRUPT FLAG
 4538  017744  012701  002446                  MOV     #L.CS,R1        ;GET ADDRESS OF LOAD REGS
 4539  017750  013711  002444                  MOV     RLDRV,(R1)      ;LOAD DRIVE NUMBER
 4540  017754  042711  002000                  BIC     #BIT10,(R1)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4541  017760  052721  000110                  BIS     #RDHEAD,(R1)+   ;INSERT COMMAND
 4542  017764  005021                          CLR     (R1)+           ;CLEAR BA
 4543  017766  005021                          CLR     (R1)+           ;CLEAR DA
 4544  017770  014162  000004                  MOV     -(R1),RLDA(R2)  ;LOAD RL11 REGS
 4545  017774  014162  000002                  MOV     -(R1),RLBA(R2)
 4546  020000  014162  000000                  MOV     -(R1),RLCSR(R2)
 4547  020004                          3$:     WAITUS  #10.            ;WAIT 1MS FOR INTERRUPT
   (3) 020004  012700  000012                  MOV     #10.,R0
   (3) 020010  104027                          EMT     C$WTU
 4548  020012  005737  002420                  TST     DONE            ;TEST IN INTERRUPT FLAG SET
 4549  020016  001455                          BEQ     14$             ;NO - SKIP
 4550  020020  032737  000001  002456  5$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 4551  020026  001033                          BNE     10$             ;YES - SKIP
 4552  020030  012703  010501                  MOV     #MDRDY,R3       ;SET NO READY MESSAGE
 4553  020034  012704  011564                  MOV     #CAFDT,R4       ;CONDITION OF AFTER DATA XFER
 4554  020040                                  ERRHRD  10017.,,ERR5
   (3) 020040  104443                          TRAP    T$ERCODE
   (5) 020042  023441                          .WORD   10017
   (5) 020044  013062                          .WORD   ERR5
 4555  020046  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 4556  020052  004737  016250          4$:     JSR     PC,GSTAT        ;GET STATUS
 4557  020056  020166                          60$
 4558  020060  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE HAS COME READY
 4559  020066  001403                          BEQ     11$             ;NO - SKIP
 4560  020070  005037  002430                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
 4561  020074  000410                          BR      10$             ;SKIP
 4562  020076  005301                  11$:    DEC     R1              ;DEC WAIT COUNT
 4563  020100  001364                          BNE     4$              ;LOOP UNTIL TIME DONE
```

```
 4564  020102  012704  011576                MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECONDS
 4565  020106                                ERRHRD  10014.,,ERR5
   (3) 020106  104443                        TRAP    T$ERCODE
   (5) 020110  023436                        .WORD   10014
   (5) 020112  013062                        .WORD   ERR5
 4566  020114  000424                        BR      60$             ;EXIT
 4567  020116  005737  002456          10$:   TST     T.CS            ;CHECK FOR ANY ERRORS
 4568  020122  100004                        BPL     12$             ;NO - SKIP
 4569  020124                                ERRHRD  10016.,,ERR6    ;REPORT ALL ERRORS
   (3) 020124  104443                        TRAP    T$ERCODE
   (5) 020126  023440                        .WORD   10016
   (5) 020130  013132                        .WORD   ERR6
 4570  020132  000415                        BR      60$
 4571  020134  012701  002466          12$:   MOV     #HDWRD2,R1      ;GET POINTER
 4572  020140  016221  000006                MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
 4573  020144  016221  000006                MOV     RLMP(R2),(R1)+
 4574  020150  000410                        BR      65$             ;EXIT
 4575  020152  004737  016050          14$:   JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 4576  020156  012603                        MOV     (SP)+,R3        ;GET RESULTS
 4577  020160                                ERRHRD  10015.,,ERR1    ;REPORT
   (3) 020160  104443                        TRAP    T$ERCODE
   (5) 020162  023437                        .WORD   10015
   (5) 020164  012630                        .WORD   ERR1
 4578  020166  005037  002430          60$:   CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 4579  020172  005737  002540          65$:   TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 4580  020176  001007                        BNE     22$             ;NO - SKIP
 4581  020200  012703  002446                MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
 4582  020204  012701  000004                MOV     #4,R1           ;SET COUNT
 4583  020210  012623                  20$:   MOV     (SP)+,(R3)+     ;RESTORE REGISTER
 4584  020212  005301                        DEC     R1              ;DEC COUNT
 4585  020214  001375                        BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
 4586  020216  162737  000002+ 002414  22$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4587  020224  012604                        MOV     (SP)+,R4        ;RESTORE REGS
 4588  020226  012601                        MOV     (SP)+,R1
 4589  020230  012600                        MOV     (SP)+,R0
 4590  020232  012603                        MOV     (SP)+,R3
 4591  020234  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
 4592  020240  001403                        BEQ     99$             ;YES - SKIP
 4593  020242  063716  002430                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4594  020246  000207                        RTS     PC
 4595  020250  017616  000000          99$:   MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4596  020254  000207                        RTS     PC
 4597
 4673                          ;
 4674  020256  013705  002464  POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
 4675  020262  000402                  BR      POSHDO          ;SKIP
 4676  020264  013705  002464  POSHSB: MOV     T.MP,R5 ;START FOR POSITION HD BIT IN MP
 4677  020270  010146          POSHDO: MOV     R1,-(SP)        ;STORE R1
 4678  020272  042705  177677          BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
 4679  020276  012701  000006          MOV     #6,R1           ;SET SHIFT COUNT
 4680  020302  006205                  1$:     ASR     R5      ;SHIFT FOR RIGHT JUSTIFY
 4681  020304  005301                  DEC     R1
 4682  020306  001375                  BNE     1$
 4683  020310  012601                  MOV     (SP)+,R1        ;RESTORE R1
 4684  020312  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  020314  010346                    RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
4689  020316  013703  002414                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
4690  020322  005723                    TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
4691  020324  016663  000002  002250    MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
4692  020332  162763  000004  002250    SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
4693  020340  010337  002414            MOV     R3,SSINDX       ;STORE IT BACK
4694  020344  010046                    MOV     R0,-(SP)
4695  020346  010146                    MOV     R1,-(SP)
4696  020350  010446                    MOV     R4,-(SP)
4697  020352  012737  000002  002430    MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
4698  020360  004737  016250            5$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
4699  020364  020516                    10$
4700  020366  032737  000001  002456    BIT     #DRDYMSK,T.CS   ;CHECK IF READY
4701  020374  001052                    BNE     9$              ;YES - SKIP
4702  020376  005301                    DEC     R1              ;DEC WAIT COUNT
4703  020400  001404                    BEQ     7$              ;SKIP IF 0
4704  020402                            WAITUS  #1
  (3) 020402  012700  000001                MOV     #1,R0
  (3) 020406  104027                    EMT     C$WTU
4705  020410  000763                    BR      5$
4706  020412  012703  010501            7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
4707  020416                            ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
  (3) 020416  104443                    TRAP    T$ERCODE
  (5) 020420  023444                    .WORD   10020
  (5) 020422  012744                    .WORD   ERR3
4708  020424  012701  000062                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
4709  020430  004737  016250            6$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
4710  020434  020516                    10$
4711  020436  032737  000001  002456    BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
4712  020444  001013                    BNE     8$              ;YES - SKIP
4713  020446                            WAITMS  #1              ;WAIT 100 MS
  (3) 020446  012700  000001                MOV     #1,R0
  (3) 020452  104026                    EMT     C$WTM
4714  020454  005301                    DEC     R1              ;DEC WAIT COUNT
4715  020456  001364                    BNE     6$              ;LOOP UNTIL TIME DONE
4716  020460  012704  011576            MOV     #C$SEC,R4       ;SET CONDITION AFTER 5 SECDS
4717  020464                            ERRHRD  10021.,,ERR5
  (3) 020464  104443                    TRAP    T$ERCODE
  (5) 020466  023445                    .WORD   10021
  (5) 020470  013062                    .WORD   ERR5
4718  020472  000407                    BR      11$             ;EXIT
4719  020474  032737  100000  002456    8$:     BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
4720  020502  001405                    BEQ     10$             ;NO - SKIP
4721  020504                            ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
  (3) 020504  104443                    TRAP    T$ERCODE
  (5) 020506  023446                    .WORD   10022
  (5) 020510  013132                    .WORD   ERR6
4722  020512  005337  002650            11$:    DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
4723  020516  005037  002430            10$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
4724  020522  162737  000002  002414    9$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
4725  020530  012604                    MOV     (SP)+,R4        ;RESTORE REGISTERS
4726  020532  012601                    MOV     (SP)+,R1
4727  020534  012600                    MOV     (SP)+,R0
4728  020536  012603                    MOV     (SP)+,R3        ;RESTORE R3
```

```
 4729  020540  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4730  020544  001403                          BEQ     99$             ;YES - SKIP
 4731  020546  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4732  020552  000207                          RTS     PC
 4733  020554  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4734  020560  000207                          RTS     PC
 4735
 4736                                  ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
 4737                                  ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
 4738                                  ;       NUMBER IN CURCYL.
 4739  020562  010346                  GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
 4740  020564  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4741  020570  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4742  020572  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4743  020600  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4744  020606  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4745  020612  010046                          MOV     R0,-(SP)
 4746  020614  010546                          MOV     R5,-(SP)
 4747  020616  004737  017636                  JSR     PC,XRDHD        ;DO READ HEADER
 4748  020622  020652                          #65$
 4749  020624  013703  002464                  MOV     HDWRD1,R3       ;GET HEADER WORD
 4750  020630  042703  100177                  BIC     #^CHDCYL,R3     ;CLEAR ALL BUT CYLINDER
 4751  020634  012705  000007                  MOV     #7,R5           ;SET SHIFT COUNT
 4752  020640  006203                  4$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
 4753  020642  005305                          DEC     R5
 4754  020644  001375                          BNE     4$
 4755  020646  010337  002516                  MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
 4756  020652  162737  000002  002414  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4757  020660  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
 4758  020662  012600                          MOV     (SP)+,R0
 4759  020664  012603                          MOV     (SP)+,R3
 4760  020666  005737  002430                  TST     ERRSWI          ;TEST IF ERROR RETURN
 4761  020672  001403                          BEQ     99$             ;YES - SKIP
 4762  020674  063716  002430                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4763  020700  000207                          RTS     PC
 4764  020702  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4765  020706  000207                          RTS     PC
 4766
 4795                                  ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
 4796                                  ;       IN IBUFF.
 4797  020710  010346                  RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
 4798  020712  013703  002414                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4799  020716  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4800  020720  016663  000002  002250          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4801  020726  162763  000004  002250          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4802  020734  010337  002414                  MOV     R3,SSINDX       ;STORE IT BACK
 4803  020740  010046                          MOV     R0,-(SP)
 4804  020742  010146                          MOV     R1,-(SP)
 4805  020744  010446                          MOV     R4,-(SP)
 4806  020746  012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4807  020754  012701  000050                  MOV     #40.,R1         ;SET HEADER COUNT
 4808  020760  052737  100000  002416          BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
 4809  020766  012703  003256                  MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
 4810  020772  013704  002440                  MOV     RLBAS,R4        ;GET BASE ADDRESS
 4811  020776  062704  000006                  ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
 4812  021002  012737  000010  002446          MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
```

```
 4813  021010  053737  002444  002446        BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
 4814  021016  042737  002000  002446        BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4815  021024  005037  002450                CLR     L.BA            ;CLEAR BA
 4816  021030  005037  002452                CLR     L.DA            ;CLEAR DA
 4817  021034  005737  002524                TST     DESHD           ;TEST IF HEAD 0
 4818  021040  001403                        BEQ     3$              ;YES - SKIP
 4819  021042  052737  000020  002452        BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
 4820  021050  013762  002452  000004  3$:   MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
 4821  021056  013762  002450  000002        MOV     L.BA,RLBA(R2)   ;LOAD RLBA
 4822  021064  032762  000200  000000        BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
 4823  021072  001003                        BNE     6$              ;YES - SKIP
 4824  021074  004737  017366                JSR     PC,RDYCHK       ;ELSE CHECK READY
 4825  021100  021212                        #65$
 4826  021102  013762  002446  000000  6$:   MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
 4827  021110  012700  077777                MOV     #77777,R0       ;SET COUNT FOR WAIT
 4828  021114  032762  000200  000000  7$:   BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
 4829  021122  001015                        BNE     8$              ;YES - SKIP
 4830  021124  005300                        DEC     R0              ;DEC COUNT
 4831  021126  001372                        BNE     7$              ;SKIP IF NOT YET 0
 4832  021130  004737  016016                JSR     PC,READRL       ;ELSE GET ALL REGISTERS
 4833  021134  004737  016050                JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
 4834  021140  012603                        MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 4835  021142                                ERRHRD  10025.,,ERR1
   (3) 021142  104443                        TRAP    T$ERCODE
   (5) 021144  023451                        .WORD   10025
   (5) 021146  012630                        .WORD   ERR1
 4836  021150  005037  002430                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4837  021154  000416                        BR      65$
 4838  021156  005737  002456        8$:     TST     T.CS            ;TEST FOR ANY ERRORS
 4839  021162  100006                        BPL     12$             ;NO - SKIP
 4840  021164                                ERRHRD  10026.,,ERR6
   (3) 021164  104443                        TRAP    T$ERCODE
   (5) 021166  023452                        .WORD   10026
   (5) 021170  013132                        .WORD   ERR6
 4841  021172  005037  002430                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 4842  021176  000405                        BR      65$
 4843  021200  011423             12$:       MOV     (R4),(R3)+      ;STORE HEADER WORDS
 4844  021202  011423                        MOV     (R4),(R3)+
 4845  021204  011423                        MOV     (R4),(R3)+
 4846  021206  005301                        DEC     R1              ;DEC HEADER COUNT
 4847  021210  001334                        BNE     6$
 4848  021212  162737  000002  002414  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 4849  021220  012604                        MOV     (SP)+,R4        ;RESTORE REGISTERS
 4850  021222  012601                        MOV     (SP)+,R1
 4851  021224  012600                        MOV     (SP)+,R0
 4852  021226  012603                        MOV     (SP)+,R3
 4853  021230  005737  002430                TST     ERRSWI          ;TEST IF ERROR RETURN
 4854  021234  001403                        BEQ     99$             ;YES - SKIP
 4855  021236  063716  002430                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 4856  021242  000207                        RTS     PC
 4857  021244  017616  000000        99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 4858  021250  000207                        RTS     PC
 4859
 4860
 5088                             ;          REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
 5089                             ;          OPERATION BEING PERFORMED PORTION OF ALL
```

```
                                ;
                                ;ERROR MESSAGES.
                        RPTOP:  MOV     R4,-(SP)
005737  002414                 TST     SSINDX          ;TEST SUBROUTINE INDEX 0
001433                          BEQ     1$              ;SKIP IF 0
012704  000002                 MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
                                PRINTB  #FMT9,#SEQMES   ;PRINT "SUBROUTINE CALL SEQ"
012746  010324                 MOV     #SEQMES,-(SP)
012746  012117                 MOV     #FMT9,-(SP)
012746  000002                 MOV     #2,-(SP)
010600                          MOV     SP,R0
104014                          EMT     C$PNTB
062706  000006                 ADD     #6,SP
                        3$:     PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
016446  002250                 MOV     SUBSTK(R4),-(SP)
012746  012272                 MOV     #FMT16,-(SP)
012746  000002                 MOV     #2,-(SP)
010600                          MOV     SP,R0
104014                          EMT     C$PNTB
062706  000006                 ADD     #6,SP
062704  000002                 ADD     #2,R4           ;BUMP INDEX
020437  002414                 CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
003761                          BLE     3$              ;LOOP IF NOT ALL PRINTED YET
                        1$:     PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
012746  006007                 MOV     #TSTLAB,-(SP)
013746  002424                 MOV     ERHEAD,-(SP)
012746  011722                 MOV     #FMT4,-(SP)
012746  000003                 MOV     #3,-(SP)
010600                          MOV     SP,R0
104014                          EMT     C$PNTB
062706  000010                 ADD     #10,SP
042737  030000  002416         BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
013701  002446                 MOV     L.CS,R1         ;GET COMMAND EXECUTED
042701  177741                 BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
022701  000006                 CMP     #6,R1   ;TEST IF SEEK OPERATION
001003                          BNE     2$              ;NO - SKIP
052737  010000  002416         BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
022701  000012         2$:     CMP     #12,R1          ;TEST IF WRITE
001003                          BNE     20$             ;NO - SKIP
052737  020000  002416         BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
022701  000014         20$:    CMP     #14,R1          ;TEST IF READ
001003                          BNE     22$             ;NO - SKIP
052737  020000  002416         BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
                        22$:    PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
016146  002112                 MOV     OPMSGS(R1),-(SP)
012746  004733                 MOV     #MOPER,-(SP)
012746  011700                 MOV     #FMT1,-(SP)
012746  000003                 MOV     #3,-(SP)
010600                          MOV     SP,R0
104014                          EMT     C$PNTB
062706  000010                 ADD     #10,SP
020127  000004                 CMP     R1,#4           ;CHECK IF GET STATUS
001007                          BNE     4$              ;NO - SKIP
032737  000010  002452         BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
001403                          BEQ     4$              ;NO - SKIP
012701  000016                 MOV     #16,R1          ;SET TO PRINT WITH RESET
000436                          BR      9$
```

```
 5120  021536  032737  007777  002416  4$:     BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
 5121  021544  001424                          BEQ     8$              ;NO - SKIP
 5122  021546  013704  002416                  MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
 5123  021552  012701  000020                  MOV     #20,R1          ;PRESET THE POINTER
 5124  021556  032704  000001          5$:     BIT     #BIT00,R4       ;CHECK THE BIT
 5125  021562  001003                          BNE     6$              ;IF SET - SKIP
 5126  021564  005721                          TST     (R1)+           ;BUMP POINTER
 5127  021566  006204                          ASR     R4
 5128  021570  000772                          BR      5$
 5129  021572                          6$:     PRINTB  #FMT2,OPMSGS(R1)
   (8) 021572  016146  002112                  MOV     OPMSGS(R1),-(SP)
   (7) 021576  012746  011714                  MOV     #FMT2,-(SP)
   (6) 021602  012746  000002                  MOV     #2,-(SP)
   (3) 021606  010600                          MOV     SP,R0
   (4) 021610  104014                          EMT     C$PNTB
   (4) 021612  062706  000006                  ADD     #6,SP
 5130  021616  032737  100000  002416  8$:     BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
 5131  021624  001415                          BEQ     10$             ;NO - SKIP
 5132  021626  012701  000050                  MOV     #50,R1          ;ELSE PRINT IT
 5133  021632                          9$:     PRINTB  #FMT2,OPMSGS(R1)
   (8) 021632  016146  002112                  MOV     OPMSGS(R1),-(SP)
   (7) 021636  012746  011714                  MOV     #FMT2,-(SP)
   (6) 021642  012746  000002                  MOV     #2,-(SP)
   (3) 021646  010600                          MOV     SP,R0
   (4) 021650  104014                          EMT     C$PNTB
   (4) 021652  062706  000006                  ADD     #6,SP
 5134  021656  000434                          BR      15$             ;SKIP
 5135  021660  032737  010000  002416  10$:    BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
 5136  021666  001430                          BEQ     15$             ;NO - SKIP
 5137  021670                                  PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
  (15) 021670  013746  002524                  MOV     DESHD,-(SP)
  (14) 021674  012746  010265                  MOV     #HDWD,-(SP)
  (13) 021700  013746  002522                  MOV     DESSGN,-(SP)
  (12) 021704  012746  010260                  MOV     #SGNWD,-(SP)
  (11) 021710  013746  002520                  MOV     DESDIF,-(SP)
  (10) 021714  012746  010252                  MOV     #DIFWD,-(SP)
   (9) 021720  013746  002512                  MOV     OLDCYL,-(SP)
   (8) 021724  012746  010303                  MOV     #FRMWD,-(SP)
   (7) 021730  012746  012140                  MOV     #FMT13,-(SP)
   (6) 021734  012746  000011                  MOV     #11,-(SP)
   (3) 021740  010600                          MOV     SP,R0
   (4) 021742  104014                          EMT     C$PNTB
   (4) 021744  062706  000024                  ADD     #24,SP
 5138  021750  032737  020000  002416  15$:    BIT     #RORWOP,OPFLAG  ;TEST IF READ OR WRITE SET
 5139  021756  001424                          BEQ     17$             ;NO - SKIP
 5140  021760                                  PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
  (13) 021760  013746  002526                  MOV     DESSEC,-(SP)
  (12) 021764  012746  010271                  MOV     #SECWD,-(SP)
  (11) 021770  013746  002524                  MOV     DESHD,-(SP)
  (10) 021774  012746  010265                  MOV     #HDWD,-(SP)
   (9) 022000  013746  002516                  MOV     CURCYL,-(SP)
   (8) 022004  012746  010276                  MOV     #CYLWD,-(SP)
   (7) 022010  012746  012467                  MOV     #FMT22,-(SP)
   (6) 022014  012746  000007                  MOV     #7,-(SP)
   (3) 022020  010600                          MOV     SP,R0
   (4) 022022  104014                          EMT     C$PNTB
```

```
   (4)  022024  062706  000020                ADD     #20,SP
 5141   022030  004737  022502        17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
 5142   022034  012604                        MOV     (SP)+,R4        ;RESTORE R4
 5143   022036  000207                        RTS     PC
 5144
 5145                                ;       REPORT REASON ROUTINE
 5146                                ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
 5147   022040  010146                RPTRES: MOV     R1,-(SP)        ;STORE R1
 5148   022042  010346                        MOV     R3,-(SP)        ;STORE R3
 5149   022044  010446                        MOV     R4,-(SP)        ;STORE R4
 5150   022046  012701  002474                MOV     #RESPARM,R1     ;GET START OF PARAM
 5151   022052  012103                        MOV     (R1)+,R3        ;GET NUMBER OF PARAM
 5152   022054                                PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
   (9)  022054  011146                        MOV     (R1),-(SP)
   (8)  022056  012746  004747                MOV     #MRSLT,-(SP)
   (7)  022062  012746  011705                MOV     #FMT1.1,-(SP)
   (6)  022066  012746  000003                MOV     #3,-(SP)
   (3)  022072  010600                        MOV     SP,R0
   (4)  022074  104014                        EMT     C$PNTB
   (4)  022076  062706  000010                ADD     #10,SP
 5153   022102  021127  011175                CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
 5154   022106  001453                        BEQ     6$              ;YES - SKIP REST OF REPORT
 5155   022110  012704  012124                MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
 5156   022114  022127  011170                CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
 5157   022120  001002                        BNE     3$              ;NO - SKIP
 5158   022122  012704  012132                MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
 5159   022126  005303                3$:     DEC     R3              ;DEC PARAM COUNT
 5160   022130  001442                        BEQ     6$              ;IF 0 - EXIT
 5161   022132                                PRINTB  R4,#RESE3,(R1)+ ;REPORT IS VALUE
   (9)  022132  012146                        MOV     (R1)+,-(SP)
   (8)  022134  012746  011437                MOV     #RESE3,-(SP)
   (7)  022140  010446                        MOV     R4,-(SP)
   (6)  022142  012746  000003                MOV     #3,-(SP)
   (3)  022146  010600                        MOV     SP,R0
   (4)  022150  104014                        EMT     C$PNTB
   (4)  022152  062706  000010                ADD     #10,SP
 5162   022156                                PRINTB  R4,#RESE4,(R1)+ ;REPORT SB VALUE
   (9)  022156  012146                        MOV     (R1)+,-(SP)
   (8)  022160  012746  011443                MOV     #RESE4,-(SP)
   (7)  022164  010446                        MOV     R4,-(SP)
   (6)  022166  012746  000003                MOV     #3,-(SP)
   (3)  022172  010600                        MOV     SP,R0
   (4)  022174  104014                        EMT     C$PNTB
   (4)  022176  062706  000010                ADD     #10,SP
 5163   022202  162703  000002                SUB     #2,R3           ;DEC PARAM COUNT
 5164   022206  001413                        BEQ     6$              ;IF 0 - EXIT
 5165   022210                                PRINTB  #FMT1,#RESE5,(R1)+      ;REPORT CONDITION
   (9)  022210  012146                        MOV     (R1)+,-(SP)
   (8)  022212  012746  011450                MOV     #RESE5,-(SP)
   (7)  022216  012746  011700                MOV     #FMT1,-(SP)
   (6)  022222  012746  000003                MOV     #3,-(SP)
   (3)  022226  010600                        MOV     SP,R0
   (4)  022230  104014                        EMT     C$PNTB
   (4)  022232  062706  000010                ADD     #10,SP
 5166   022236  012604                6$:     MOV     (SP)+,R4        ;RESTORE REGS
 5167   022240  012603                        MOV     (SP)+,R3
```

```
                                                MOV     (SP)+,R1
                                                RTS     PC                      ;RETURN

                                        ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
                                        ;       AND ALL REGISTER CONTENTS.
                                        RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
                                                CLR     -(SP)
                                                BISB    RLDRV+1,(SP)
                                                MOV     #DRVNAM,-(SP)
                                                MOV     RLBAS,-(SP)
                                                MOV     #BASADD,-(SP)
                                                MOV     #FMT5,-(SP)
                                                MOV     #5,-(SP)
                                                MOV     SP,R0
                                                EMT     C$PNTB
                                                ADD     #14,SP
                                        ;       REPORT RL11 REGISTERS
                                                PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
                                                MOV     #HDWD,-(SP)
                                                MOV     #CYLWD,-(SP)
                                                MOV     #MPNAM,-(SP)
                                                MOV     #BANAM,-(SP)
                                                MOV     #DANAM,-(SP)
                                                MOV     #CSNAM,-(SP)
                                                MOV     #FMT6,-(SP)
                                                MOV     #7,-(SP)
                                                MOV     SP,R0
                                                EMT     C$PNTB
                                                ADD     #20,SP
                                                PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
                                                MOV     L.MP,-(SP)
                                                MOV     L.BA,-(SP)
                                                MOV     L.DA,-(SP)
                                                MOV     L.CS,-(SP)
                                                MOV     #LAB1,-(SP)
                                                MOV     #FMT8,-(SP)
                                                MOV     #6,-(SP)
                                                MOV     SP,R0
                                                EMT     C$PNTB
                                                ADD     #16,SP
                                                PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
                                                MOV     DESHD,-(SP)
                                                MOV     CURCYL,-(SP)
                                                MOV     T.MP,-(SP)
                                                MOV     T.BA,-(SP)
                                                MOV     T.DA,-(SP)
                                                MOV     T.CS,-(SP)
                                                MOV     #LAB2,-(SP)
                                                MOV     #FMT7,-(SP)
                                                MOV     #10,-(SP)
                                                MOV     SP,R0
                                                EMT     C$PNTB
                                                ADD     #22,SP
                                                RTS     PC

                                        ;       CLEAR PARAMETER BLOCK FOR REPORTING
```

```
 5181   022502   010546                 CLRPARM:        MOV     R5,-(SP)        ;STORE R5
 5182   022504   012701   002474                        MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
 5183   022510   012705   000005                        MOV     #5,R5           ;SET COUNT
 5184   022514   005021                 2$:             CLR     (R1)+           ;CLEAR WORD
 5185   022516   005305                                 DEC     R5              ;DEC COUNT
 5186   022520   001375                                 BNE     2$              ;LOOP UNTIL 0
 5187   022522   012701   002474                        MOV     #RESPARM,R1     ;RESET POINTER
 5188   022526   012605                                 MOV     (SP)+,R5        ;RESTORE R5
 5189   022530   000207                                 RTS     PC
 5190
 5191   022532                         ENDMOD
 5192
```

```
 5194  022532                          BGNMOD  HRDWTST
 5195                                  .SBTTL  *TEST 1          BASIC INTERFACE (PART 1)
 5196
 5197  022532                          BGNTST                   ;TEST01
   (3) 022532                                                                  T1::
 5198  022532  005737  002652                  TST     PASNUM           ;CHECK IF FIRST PASS
 5199  022536  001120                          BNE     65$              ;EXIT IF NO
 5200  022540  005737  014446                  TST     MISWIW           ;CHECK IF MANUAL INTERVENTION
 5201  022544  100115                          BPL     65$              ;NO - EXIT TEST
 5202  022546  012737  006015  002424          MOV     #MISTST,ERHEAD   ;LOAD ERR HEADER
 5203  022554                          2$:     PRINTF  #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (13) 022554  005046                          CLR     -(SP)
  (13) 022556  153716  002445                  BISB    RLDRV+1,(SP)
  (12) 022562  012746  005423                  MOV     #DRVNAM,-(SP)
  (11) 022566  013746  002440                  MOV     RLBAS,-(SP)
  (10) 022572  012746  005412                  MOV     #BASADD,-(SP)
   (9) 022576  012746  010206                  MOV     #OPR1A,-(SP)
   (8) 022602  012746  007564                  MOV     #OPR1,-(SP)
   (7) 022606  012746  011606                  MOV     #FMTOP1,-(SP)
   (6) 022612  012746  000007                  MOV     #7,-(SP)
   (3) 022616  010600                          MOV     SP,R0
   (4) 022620  104017                          EMT     C$PNTF
   (4) 022622  062706  000020                  ADD     #20,SP
 5204  022626  005037  003656                  CLR     OBUFF            ;CLEAR FOR RESPONSE
 5205  022632                                  GMANIL  #OPR002,OBUFF,1,NO
   (3) 022632  104043                          EMT     C$GMAN
   (3) 022634  000404                          BR      10000$
   (4) 022636  003656                          .WORD   OBUFF
   (5) 022640  000120                          .WORD   T$CODE
   (5) 022642  007514                          .WORD   #OPR002
   (5) 022644  000001                          .WORD   1
   (3) 022646                          10000$:
 5206  022646  005737  003656                  TST     OBUFF            ;TEST RESPONSE YES
 5207  022652  001740                          BEQ     2$               ;YES - SKIP
 5208  022654  004737  016202          1$:     JSR     PC,TSTINT                ;INITIALIZE TEST
 5209  022660  004737  016234                  JSR     PC,GSTATC        ;GO GET STATUS (NO RESET)
 5210  022664  023000                          #65$
 5211  022666  032737  000040  002464          BIT     #COSTAT,T.MP     ;CHECK IF COVER OPEN SET
 5212  022674  001005                          BNE     7$               ;YES - SKIP
 5213  022676  012703  010662                  MOV     #MCOSTA,R3       ;SET NAME POINTER
 5214  022702                                  ERRHRD  101,,ERR3
   (3) 022702  104443                          TRAP    T$ERCODE
   (5) 022704  000145                          .WORD   101
   (5) 022706  012744                          .WORD   ERR3
 5215  022710  032737  000010  002464  7$:     BIT     #BHSTAT,T.MP     ;TEST IF BRUSHES HOME
 5216  022716  001005                          BNE     9$               ;YES - SKIP
 5217  022720  012703  010675                  MOV     #MBHSTA,R3       ;SET POINTER FOR BRUSH HOME ERROR
 5218  022724                                  ERRHRD  102,,ERR3
   (3) 022724  104443                          TRAP    T$ERCODE
   (5) 022726  000146                          .WORD   102
   (5) 022730  012744                          .WORD   ERR3
 5219  022732  032737  020000  002464  9$:     BIT     #WLSTAT,T.MP     ;TEST IF WRITE LOCK SET
 5220  022740  001005                          BNE     11$              ;YES - SKIP
 5221  022742  012703  010710                  MOV     #MWLSTA,R3       ;SET NAME POINTER
 5222  022746                                  ERRHRD  103,,ERR3
   (3) 022746  104443                          TRAP    T$ERCODE
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-1
DZRLCA.P11    05-OCT-77 10:52          *TEST 1          BASIC INTERFACE (PART 1)                    SEQ 0075

   (5)  022750  000147                          .WORD   103
   (5)  022752  012744                          .WORD   ERR3
 5223   022754  005737  002472          11$:    TST     T.STAT          ;TEST IF STATE ZERO
 5224   022760  001404                          BEQ     15$             ;YES - SKIP
 5225   022762  005003                          CLR     R3              ;SET STATE EXPECTED
 5226   022764                                  ERRHRD  104,,ERR7
   (3)  022764  104443                          TRAP    T$ERCODE
   (5)  022766  000150                          .WORD   104
   (5)  022770  014010                          .WORD   ERR7
 5227   022772  004737  016220          15$:    JSR     PC,G$TATR       ;DO DRIVE RESET
 5228   022776  023000                          65$
 5229   023000                          65$:
 5230   023000                          ENDTST
   (3)  023000                          L10020:
   (3)  023000  104001                          EMT     C$ETST
 5231
```

```
                                        .SBTTL  *TEST 2          BASIC INTERFACE (PART 2)

                                        BGNTST                   ;TEST 2
                                                                                        T2::
                                             TST     PASNUM              ;TEST IF PASS 0
                                             BNE     65$                 ;NO - SKIP
                                             TST     MISWIW              ;TEST IF MANUAL INTERVENTION
                                             BPL     65$                 ;NO - SKIP
                                             MOV     #MISTST,ERHEAD      ;SET ERROR HEADER
                                        2$:  PRINTF  #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>  ;REQUEST CLOSE
                                             CLR     -(SP)
                                             BISB    RLDRV+1,(SP)
                                             MOV     #DRVNAM,-(SP)
                                             MOV     RLBAS,-(SP)
                                             MOV     #BASADD,-(SP)
                                             MOV     #OPR1A,-(SP)
                                             MOV     #OPR2,-(SP)
                                             MOV     #FMTOP1,-(SP)
                                             MOV     #7,-(SP)
                                             MOV     SP,R0
                                             EMT     C$PNTF
                                             ADD     #20,SP
                                                                 ;COVER AND RESET WRITE LOCK
                                             CLR     OBUFF       ;CLEAR FOR RESPONSE
                                             GMANIL  #OPR002,OBUFF,1,NO
                                             EMT     C$GMAN
                                             BR      10000$
                                             .WORD   OBUFF
                                             .WORD   T$CODE
                                             .WORD   #OPR002
                                             .WORD   1
                                        10000$:
                                             TST     OBUFF               ;TEST IF RESPONSE YES
                                             BEQ     2$                  ;NO - SKIP
                                        1$:  JSR     PC,TSTINT                   ;INITIALIZE TEST
                                             JSR     PC,GSTATR           ;GET STATUS WITH RESET
                                             #65$
                                             BIT     #COSTAT,T.MP        ;TEST IF COVER OPEN RESET
                                             BEQ     9$                  ;YES - SKIP
                                             MOV     #MCOSTA,R3          ;SET NAME MESSAGE POINTER
                                             ERRHRD  201,,ERR2
                                             TRAP    T$ERCODE
                                             .WORD   201
                                             .WORD   ERR2
                                        9$:  BIT     #WLSTAT,T.MP        ;TEST IF WRITE LOCK RESET
                                             BEQ     65$                 ;YES - SKIP
                                             MOV     #MWLSTA,R3          ;SET NAME MESSAGE POINTER
                                             ERRHRD  202,,ERR2
                                             TRAP    T$ERCODE
                                             .WORD   202
                                             .WORD   ERR2
                                        65$:
                                        ENDTST
                                        L10021:
                                             EMT     C$ETST
```

```
                                .SBTTL  *TEST 3          HEAD LOADING
                        BGNTST                           ;TEST03
                                                                        T3::
                                TST     PASNUM           ;TEST IF PASS 0
                                BNE     4$               ;NO SKIP
                                TST     MISWIW           ;TEST IF MANUAL INTERVENTION
                                BMI     5$               ;YES - SKIP
                        4$:     EXIT    TST
                                EMT     C$EXIT
                                .WORD   L10022-.
                        5$:     JSR     PC,TSTINT                ;INITIALIZE TEST
                                JSR     PC,GSTATR        ;GET STATUS
                                #T365$
                                TST     T.STAT           ;TEST IF STATE ZERO
                                BEQ     2$               ;YES - SKIP
                        1$:     PRINTF  #FMTOP1,#OPR5,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST DRIVE BE
                                CLR     -(SP)
                                BISB    RLDRV+1,(SP)
                                MOV     #DRVNAM,-(SP)
                                MOV     RLBAS,-(SP)
                                MOV     #BASADD,-(SP)
                                MOV     #OPR1A,-(SP)
                                MOV     #OPR5,-(SP)
                                MOV     #FMTOP1,-(SP)
                                MOV     #7,-(SP)
                                MOV     SP,R0
                                EMT     C$PNTF
                                ADD     #20,SP
                                CLR     OBUFF            ;CLEAR FOR RESPONSE
                                GMANIL  #OPR002,OBUFF,1,NO
                                EMT     C$GMAN
                                BR      10000$
                                .WORD   OBUFF
                                .WORD   T$CODE
                                .WORD   #OPR002
                                .WORD   1
                        10000$: TST     OBUFF            ;TEST IF RESPONSE YES
                                BEQ     1$               ;NO - SKIP
                        2$:     PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
                                CLR     -(SP)
                                BISB    RLDRV+1,(SP)
                                MOV     #DRVNAM,-(SP)
                                MOV     RLBAS,-(SP)
                                MOV     #BASADD,-(SP)
                                MOV     #OPR1A,-(SP)
                                MOV     #OPR3,-(SP)
                                MOV     #FMTOP1,-(SP)
                                MOV     #7,-(SP)
                                MOV     SP,R0
                                EMT     C$PNTF
                                ADD     #20,SP
                                MOV     #CYLUP,OPFLAG    ;SET CYCLE UP FLAG
                                MOV     #1,R3            ;SET EXPECTED STATE VALUE
                                MOV     #NSTACHG,ERHEAD  ;SET ERROR HEADER
                                MOV     #300.,R1         ;SET WAIT COUNT FOR 30 SECONDS
```

```
 5284  023442  004737  016234          3$:     JSR     PC,GSTATC       ;GET STATUS
 5285  023446  024436                          #T365$
 5286  023450  005737  002472                  TST     T.STAT          ;TEST IF STATE IS STILL 0
 5287  023454  001022                          BNE     10$             ;NO - SKIP
 5288  023456  005301                          DEC     R1              ;DEC WAIT COUNT
 5289  023460  001404                          BEQ     6$              ;EXIT IF WAIT DONE
 5290  023462                                  WAITMS  #1
   (3) 023462  012700  000001                  MOV     #1,R0
   (3) 023466  104026                          EMT     C$WTM
 5291  023470  000764                          BR      3$
 5292  023472  005037  003656          6$:     CLR     OBUFF           ;CLEAR FOR RESPONSE
 5293  023476                                  GMANIL  #OPR003,OBUFF,1,NO
   (3) 023476  104043                          EMT     C$GMAN
   (3) 023500  000404                          BR      10001$
   (4) 023502  003656                          .WORD   OBUFF
   (5) 023504  000120                          .WORD   T$CODE
   (5) 023506  007541                          .WORD   #OPR003
   (5) 023510  000001                          .WORD   1
   (3) 023512                          10001$:
 5294  023512  005737  003656                  TST     OBUFF           ;TEST IF RESPONSE YES
 5295  023516  001004                          BNE     11$             ;YES - REPORT
 5296  023520  000651                          BR      1$
 5297  023522  020337  002472          10$:    CMP     R3,T.STAT       ;CHECK IF NOW STATE 1
 5298  023526  001405                          BEQ     13$             ;YES - SKIP
 5299  023530                          11$:    ERRHRD  301,,ERR7
   (3) 023530  104443                          TRAP    T$ERCODE
   (5) 023532  000455                          .WORD   301
   (5) 023534  014010                          .WORD   ERR7
 5300  023536                                  EXIT    TST
   (3) 023536  104032                          EMT     C$EXIT
   (3) 023540  000676                          .WORD   L10022-.
 5301  023542  012701  000454          13$:    MOV     #300.,R1        ;SET WAIT FOR 30 SECONDS
 5302  023546  012703  000002                  MOV     #2,R3           ;SET EXPECTED STATE VALUE
 5303  023552  004737  016234          14$:    JSR     PC,GSTATC       ;GET STATUS
 5304  023556  024436                          #T365$
 5305  023560  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE 2
 5306  023564  001435                          BEQ     20$             ;YES - SKIP
 5307  023566  101005                          BHI     17$             ;CHECK IF NO CHANGE - YES - SKIP
 5308  023570                                  ERRHRD  302,,ERR7
   (3) 023570  104443                          TRAP    T$ERCODE
   (5) 023572  000456                          .WORD   302
   (5) 023574  014010                          .WORD   ERR7
 5309  023576                                  EXIT    TST
   (3) 023576  104032                          EMT     C$EXIT
   (3) 023600  000636                          .WORD   L10022-.
 5310  023602  005301          17$:    DEC     R1              ;DEC WAIT COUNT
 5311  023604  001404                          BEQ     18$             ;SKIP IF 0
 5312  023606                                  WAITMS  #1
   (3) 023606  012700  000001                  MOV     #1,R0
   (3) 023612  104026                          EMT     C$WTM
 5313  023614  000756                          BR      14$
 5314  023616                          18$:    ERRHRD  303,,ERR7
   (3) 023616  104443                          TRAP    T$ERCODE
   (5) 023620  000457                          .WORD   303
   (5) 023622  014010                          .WORD   ERR7
 5315  023624  032737  004000  002464          BIT     #SPDSTAT,T.MP   ;TEST IF SPINDLE TIMEOUT
```

```
 5316  023632  001010                        BNE     19$                     ;YES - SKIP
 5317  023634  012737  006052  002424        MOV     #SPDERR,ERHEAD          ;SET ERROR HEADER
 5318  023642  012703  010762                MOV     #MSPERR,R3              ;SET NAME MESSAGE POINTER
 5319  023646                                ERRHRD  304.,,ERR3
   (3) 023646  104443                        TRAP    T$ERCODE
   (5) 023650  000460                        .WORD   304
   (5) 023652  012744                        .WORD   ERR3
 5320  023654                          19$:  EXIT    TST
   (3) 023654  104032                        EMT     C$EXIT
   (3) 023656  000560                        .WORD   L10022-.
 5321  023660  012737  006015  002424  20$:  MOV     #MISTST,ERHEAD          ;SET ERROR HEADER
 5322  023666  012704  011462                MOV     #STATE2,R4              ;SET CONDITION MESSAGE POINTER
 5323  023672  012703  010675                MOV     #MBHSTA,R3              ;SET NAME MESSAGE POINTER
 5324  023676  032737  000010  002464        BIT     #BHSTAT,T.MP            ;TEST IF BRUSH HOME STILL SET
 5325  023704  001005                        BNE     22$                     ;YES - SKIP
 5326  023706                                ERRHRD  305.,,ERR5
   (3) 023706  104443                        TRAP    T$ERCODE
   (5) 023710  000461                        .WORD   305
   (5) 023712  013062                        .WORD   ERR5
 5327  023714                                EXIT    TST
   (3) 023714  104032                        EMT     C$EXIT
   (3) 023716  000520                        .WORD   L10022-.
 5328  023720  012701  000062          22$:  MOV     #50.,R1                 ;SET WAIT COUNT FOR 5 SECONDS
 5329  023724  004737  016234          23$:  JSR     PC,GSTATC               ;GET STATUS
 5330  023730  024436                        #T365$
 5331  023732  032737  000010  002464        BIT     #BHSTAT,T.MP            ;TEST IF BRUSH HOME RESET
 5332  023740  001413                        BEQ     27$                     ;YES - SKIP
 5333  023742  005301                        DEC     R1                      ;DEC WAIT COUNT
 5334  023744  001404                        BEQ     26$                     ;SKIP IF ZERO
 5335  023746                                WAITMS  #1
   (3) 023746  012700  000001                MOV     #1,R0
   (3) 023752  104026                        EMT     C$WTM
 5336  023754  000763                        BR      23$                     ;LOOP
 5337  023756                          26$:  ERRHRD  306.,,ERR4
   (3) 023756  104443                        TRAP    T$ERCODE
   (5) 023760  000462                        .WORD   306
   (5) 023762  013012                        .WORD   ERR4
 5338  023764                                EXIT    TST
   (3) 023764  104032                        EMT     C$EXIT
   (3) 023766  000450                        .WORD   L10022-.
 5339  023770  012701  000454          27$:  MOV     #300.,R1                ;SET WAIT COUNT 30 SECONDS
 5340  023774  004737  016234          28$:  JSR     PC,GSTATC               ;GET STATUS
 5341  024000  024436                        #T365$
 5342  024002  032737  000010  002464        BIT     #BHSTAT,T.MP            ;TEST IF BRUSH HOME SET AGAIN
 5343  024010  001013                        BNE     32$                     ;YES - SKIP
 5344  024012  005301                        DEC     R1                      ;ELSE DEC WAIT COUNT
 5345  024014  001404                        BEQ     30$                     ;SKIP IF 0
 5346  024016                                WAITMS  #1
   (3) 024016  012700  000001                MOV     #1,R0
   (3) 024022  104026                        EMT     C$WTM
 5347  024024  000763                        BR      28$
 5348  024026                          30$:  ERRHRD  307.,,ERR5
   (3) 024026  104443                        TRAP    T$ERCODE
   (5) 024030  000463                        .WORD   307
   (5) 024032  013062                        .WORD   ERR5
 5349  024034                                EXIT    TST
```

```
                (3)  024034  104032                        EMT     C$EXIT
                (3)  024036  000400                        .WORD   L10022-.
5350            024040  012737  006040  002424  32$:      MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
5351            024046  012703  000003                     MOV     #3,R3           ;SET EXPECTED STATE VALUE
5352            024052  004737  016234                     JSR     PC,GSTATC       ;GET STATUS
5353            024056  024436                             T365$
5354            024060  020337  002472                     CMP     R3,T.STAT       ;CHECK IF STATE 3
5355            024064  001405                             BEQ     36$             ;YES - SKIP
5356            024066                                     ERRHRD  308.,,ERR7
                (3)  024066  104443                        TRAP    T$ERCODE
                (5)  024070  000464                        .WORD   308
                (5)  024072  014010                        .WORD   ERR7
5357            024074                                     EXIT    TST
                (3)  024074  104032                        EMT     C$EXIT
                (3)  024076  000340                        .WORD   L10022-.
5358            024100  012737  006015  002424  36$:      MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
5359            024106  012704  011472                     MOV     #STATE3,R4      ;SET CONDITION MESSAGE POINTER
5360            024112  012703  010721                     MOV     #MHOSTA,R3      ;SET NAME MESSAGE POINTER
5361            024116  004737  016234                     JSR     PC,GSTATC       ;GET STATUS
5362            024122  024436                             #T365$
5363            024124  032737  000020  002464           BIT     #HOSTAT,T.MP    ;TEST IF HEADS OUT SET
5364            024132  001005                             BNE     38$             ;YES - SKIP
5365            024134                                     ERRHRD  309.,,ERR5
                (3)  024134  104443                        TRAP    T$ERCODE
                (5)  024136  000465                        .WORD   309
                (5)  024140  013062                        .WORD   ERR5
5366            024142                                     EXIT    TST
                (3)  024142  104032                        EMT     C$EXIT
                (3)  024144  000272                        .WORD   L10022-.
5367            024146  032737  001000  002464  38$:      BIT     #VCSTAT,T.MP    ;TEST IF VOLUME CHECK SET
5368            024154  001007                             BNE     40$
5369            024156  012703  010651                     MOV     #MVOLCK,R3      ;SET NAME MESSAGE POINTER
5370            024162                                     ERRHRD  310.,,ERR5
                (3)  024162  104443                        TRAP    T$ERCODE
                (5)  024164  000466                        .WORD   310
                (5)  024166  013062                        .WORD   ERR5
5371            024170                                     EXIT    TST
                (3)  024170  104032                        EMT     C$EXIT
                (3)  024172  000244                        .WORD   L10022-.
5372            024174  032737  040000  002456  40$:      BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR SET
5373            024202  001007                             BNE     42$             ;YES - SKIP
5374            024204  012703  010623                     MOV     #MDRERR,R3      ;SET NAME MESSAGE POINTER
5375            024210                                     ERRHRD  311.,,ERR5
                (3)  024210  104443                        TRAP    T$ERCODE
                (5)  024212  000467                        .WORD   311
                (5)  024214  013062                        .WORD   ERR5
5376            024216                                     EXIT    TST
                (3)  024216  104032                        EMT     C$EXIT
                (3)  024220  000216                        .WORD   L10022-.
5377            024222  012701  005670           42$:      MOV     #3000.,R1               ;SET WAIT COUNT FOR 300 MS
5378            024226  012737  006040  002424           MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
5379            024234  012703  000004                     MOV     #4,R3           ;SET EXPECTED STATE VALUE
5380            024240  004737  016234           43$:      JSR     PC,GSTATC       ;GET STATUS
5381            024244  024436                             #T365$
5382            024246  020337  002472                     CMP     R3,T.STAT       ;CHECK IF STATE 4
5383            024252  001413                             BEQ     49$             ;YES - SKIP
```

```
 5384  024254  005301                          DEC     R1              ;DEC WAIT COUNT
 5385  024256  001404                          BEQ     47$             ;SKIP IF 0
 5386  024260                                  WAITUS  #1
   (3) 024260  012700  000001                  MOV     #1,R0
   (3) 024264  104027                          EMT     C$WTU
 5387  024266  000764                          BR      43$
 5388  024270                          47$:    ERRHRD  312.,,ERR7
   (3) 024270  104443                          TRAP    T$ERCODE
   (5) 024272  000470                          .WORD   312
   (5) 024274  014010                          .WORD   ERR7
 5389  024276                                  EXIT    TST
   (3) 024276  104032                          EMT     C$EXIT
   (3) 024300  000136                          .WORD   L10022-.
 5390  024302  012701  000454          49$:    MOV     #300.,R1        ;SET WAIT COUNT FOR 30 MS
 5391  024306  012703  000005                  MOV     #5,R3           ;SET EXPECTED STATE VALUE
 5392  024312  004737  016234          50$:    JSR     PC,GSTATC       ;GET STATUS
 5393  024316  024436                          #T365$
 5394  024320  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE 5
 5395  024324  001413                          BEQ     55$             ;YES - SKIP
 5396  024326  005301                          DEC     R1              ;DEC WAIT COUNT
 5397  024330  001404                          BEQ     51$             ;ELSE SKIP
 5398  024332                                  WAITUS  #1
   (3) 024332  012700  000001                  MOV     #1,R0
   (3) 024336  104027                          EMT     C$WTU
 5399  024340  000764                          BR      50$
 5400  024342                          51$:    ERRHRD  313.,,ERR7
   (3) 024342  104443                          TRAP    T$ERCODE
   (5) 024344  000471                          .WORD   313
   (5) 024346  014010                          .WORD   ERR7
 5401  024350                                  EXIT    TST
   (3) 024350  104032                          EMT     C$EXIT
   (3) 024352  000064                          .WORD   L10022-.
 5402  024354  012701  000120          55$:    MOV     #80.,R1         ;SET WAIT FOR 8 MS
 5403  024360  004737  016234          56$:    JSR     PC,GSTATC       ;GET STATUS
 5404  024364  024436                          #T365$
 5405  024366  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;CHECK IF DRIVE READY
 5406  024374  001020                          BNE     62$             ;YES - SKIP
 5407  024376  005301                          DEC     R1              ;DEC COUNT
 5408  024400  001404                          BEQ     60$             ;SKIP IF 0
 5409  024402                                  WAITUS  #1
   (3) 024402  012700  000001                  MOV     #1,R0
   (3) 024406  104027                          EMT     C$WTU
 5410  024410  000763                          BR      56$
 5411  024412  012737  006015  002424  60$:    MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
 5412  024420  012704  011502                  MOV     #STATES,R4      ;SET CONDITION MESSAGE POINTER
 5413  024424  012703  010501                  MOV     #MDRDY,R3       ;SET NAME MESSAGE POINTER
 5414  024430                                  ERRHRD  314.,,ERR5
   (3) 024430  104443                          TRAP    T$ERCODE
   (5) 024432  000472                          .WORD   314
   (5) 024434  013062                          .WORD   ERR5
 5415
 5416  024436                          62$:
 5417  024436                          T365$:
 5418  024436                          ENDTST
   (3) 024436                          L10022:
   (3) 024436  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 4          HEAD UNLOADING
5421    024440                          BGNTST                   ;TEST04
  (3)   024440                                                                  T4::
5422    024440  005737  002652                  TST     PASNUM           ;TEST IF FIRST PASS
5423    024444  001003                          BNE     8$               ;NO - SKIP
5424    024446  005737  014446                  TST     MISWIW           ;TEST IF MANUAL INTERVENTION
5425    024452  100402                          BMI     10$              ;YES - SKIP
5426    024454                          8$:     EXIT TST
  (3)   024454  104032                          EMT     C$EXIT
  (3)   024456  000566                          .WORD   L10023-.
5427    024460                          10$:
5428    024460                          BGNSUB
  (3)   024460                                                                  T4.1:
  (3)   024460  104002                          EMT     C$BSUB
5429    024462  012737  006040  002424          MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
5430    024470  004737  016202                  JSR     PC,TSTINT                ;INITIALIZE TEST
5431    024474  004737  016220                  JSR     PC,GSTATR        ;GET STATUS
5432    024500  025134                          #T465$
5433    024502  032737  000001  002456          BIT     #DRDYMSK,T.CS    ;CHECK IF DRIVE READY
5434    024510  001040                          BNE     3$               ;YES - SKIP
5435    024512                          1$:     PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (13)  024512  005046                          CLR     -(SP)
  (13)  024514  153716  002445                  BISB    RLDRV+1,(SP)
  (12)  024520  012746  005423                  MOV     #DRVNAM,-(SP)
  (11)  024524  013746  002440                  MOV     RLBAS,-(SP)
  (10)  024530  012746  005412                  MOV     #BASADD,-(SP)
  (9)   024534  012746  010206                  MOV     #OPR1A,-(SP)
  (8)   024540  012746  007752                  MOV     #OPR6,-(SP)
  (7)   024544  012746  011606                  MOV     #FMTOP1,-(SP)
  (6)   024550  012746  000007                  MOV     #7,-(SP)
  (3)   024554  010600                          MOV     SP,R0
  (4)   024556  104017                          EMT     C$PNTF
  (4)   024560  062706  000020                  ADD     #20,SP
5436    024564  005037  003656                  CLR     OBUFF            ;CLEAR FOR RESPONSE
5437    024570                                  GMANIL  #OPR002,OBUFF,1,NO
  (3)   024570  104043                          EMT     C$GMAN
  (3)   024572  000404                          BR      10000$
  (4)   024574  003656                          .WORD   OBUFF
  (5)   024576  000120                          .WORD   T$CODE
  (5)   024600  007514                          .WORD   #OPR002
  (5)   024602  000001                          .WORD   1
  (3)   024604                          10000$:
5438    024604  005737  003656                  TST     OBUFF            ;TST RESPONSE YES
5439    024610  001740                          BEQ     1$               ;NO - SKIP
5440
5441    024612  052737  000010  002416  3$:     BIS     #ULOAD,OPFLAG    ;SET UNLOAD OPERATION
5442    024620                          4$:     PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (13)  024620  005046                          CLR     -(SP)
  (13)  024622  153716  002445                  BISB    RLDRV+1,(SP)
  (12)  024626  012746  005423                  MOV     #DRVNAM,-(SP)
  (11)  024632  013746  002440                  MOV     RLBAS,-(SP)
  (10)  024636  012746  005412                  MOV     #BASADD,-(SP)
  (9)   024642  012746  010206                  MOV     #OPR1A,-(SP)
  (8)   024646  012746  007674                  MOV     #OPR3,-(SP)
  (7)   024652  012746  011606                  MOV     #FMTOP1,-(SP)
  (6)   024656  012746  000007                  MOV     #7,-(SP)
```

```
   (3)  024662  010600                          MOV     SP,R0
   (4)  024664  104017                          EMT     C$PNTF
   (4)  024666  062706  000020                  ADD     #20,SP
 5443   024672  012703  000006                  MOV     #6,R3           ;SET EXPECTED STATE VALUE
 5444   024676  012704  000144                  MOV     #100.,R4        ;SET SECOND LEVEL COUNT
 5445   024702  012701  001274                  MOV     #700.,R1        ;SET WAIT COUNT FOR 30 SECONDS
 5446   024706  004737  016234          5$:     JSR     PC,GSTATC       ;GET STATUS
 5447   024712  025134                          T465$
 5448   024714  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE 6
 5449   024720  001436                          BEQ     11$             ;YES - SKIP
 5450   024722  022737  000005  002472          CMP     #5,T.STAT       ;TEST IF STATE 5
 5451   024730  001025                          BNE     9$              ;NO - REPORT WRONG STATE
 5452   024732  005304                  8$:     DEC     R4              ;DEC 2ND LEVEL COUNT
 5453   024734  001004                          BNE     6$              ;SKIP IF NOT 0
 5454   024736  005301                          DEC     R1              ;ELSE DEC 1ST LEVEL COUNT
 5455   024740  001406                          BEQ     7$              ;IF 0 - SKIP TO QUESTION
 5456   024742  012704  000144                  MOV     #100.,R4        ;ELSE RESET 2ND LEVEL
 5457   024746                          6$:     WAITUS  #1              ;WAIT 100 US
   (3)  024746  012700  000001                  MOV     #1,R0
   (3)  024752  104027                          EMT     C$WTU
 5458   024754  000754                          BR      5$
 5459   024756  005037  003656          7$:     CLR     OBUFF           ;CLEAR FOR RESPONSE
 5460   024762                                  GMANIL  #OPR003,OBUFF,1,NO
   (3)  024762  104043                          EMT     C$GMAN
   (3)  024764  000404                          BR      10001$
   (4)  024766  003656                          .WORD   OBUFF
   (5)  024770  000120                          .WORD   T$CODE
   (5)  024772  007541                          .WORD   #OPR003
   (5)  024774  000001                          .WORD   1
   (3)  024776                          10001$:
 5461   024776  005737  003656                  TST     OBUFF           ;TEST IF RESPONSE YES
 5462   025002  001706                          BEQ     4$              ;NO - SKIP
 5463   025004                          9$:     ERRHRD  401.,,ERR7      ;ELSE REPORT STATE CHANGE WRONG
   (3)  025004  104443                          TRAP    T$ERCODE
   (5)  025006  000621                          .WORD   401
   (5)  025010  014010                          .WORD   ERR7
 5464   025012                                  EXIT    SUB
   (3)  025012  104032                          EMT     C$EXIT
   (3)  025014  000126                          .WORD   L10024-.
 5465   025016  012703  000007          11$:    MOV     #7,R3           ;SET EXPECTED STATE VALUE
 5466   025022  012701  005670                  MOV     #3000.,R1       ;SET COUNT FOR 300MS
 5467   025026  004737  016234          12$:    JSR     PC,GSTATC       ;GET STATUS
 5468   025032  025134                          #T465$
 5469   025034  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE 7
 5470   025040  001413                          BEQ     18$             ;YES - SKIP
 5471   025042  005301                          DEC     R1              ;DEC WAIT COUNT
 5472   025044  001404                          BEQ     16$             ;SKIP IF 0
 5473   025046                                  WAITUS  #1
   (3)  025046  012700  000001                  MOV     #1,R0
   (3)  025052  104027                          EMT     C$WTU
 5474   025054  000764                          BR      12$
 5475   025056                          16$:    ERRHRD  402.,,ERR7      ;REPORT WRONG STATE CHANGE
   (3)  025056  104443                          TRAP    T$ERCODE
   (5)  025060  000622                          .WORD   402
   (5)  025062  014010                          .WORD   ERR7
 5476   025064                                  EXIT    SUB
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-10
DZRLCA.P11    05-OCT-77 10:52          *TEST 4          HEAD UNLOADING                                                    SEQ 0084

   (3)  025064  104032                          EMT     C$EXIT
   (3)  025066  000054                          .WORD   L10024-.
 5477   025070  005003                  18$:    CLR     R3              ;SET EXPECTED STATE VALUE
 5478   025072  012701  001130                  MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
 5479   025076  004737  016234          20$:    JSR     PC,GSTATC       ;GET STATUS
 5480   025102  025134                          #T465$
 5481   025104  005737  002472                  TST     T.STAT          ;CHECK IF STATE 0
 5482   025110  001411                          BEQ     24$             ;YES - SKIP
 5483   025112  005301                          DEC     R1              ;DEC WAIT COUNT
 5484   025114  001404                          BEQ     22$             ;SKIP IF 0
 5485   025116                                  WAITMS  #1
   (3)  025116  012700  000001                  MOV     #1,R0
   (3)  025122  104026                          EMT     C$WTM
 5486   025124  000764                          BR      20$
 5487   025126                          22$:    ERRHRD  403,,ERR7       ;REPORT WRONG STATE CHANGE
   (3)  025126  104443                          TRAP    T$ERCODE
   (5)  025130  000623                          .WORD   403
   (5)  025132  014010                          .WORD   ERR7
 5488   025134                          24$:
 5489   025134  012737  000002  002430  T465$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 5490
 5491   025142                          ENDSUB
   (3)  025142                          L10024:
   (3)  025142  104003                          EMT     C$ESUB
 5492   025144                          26$:    PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CYCLE UP
  (13)  025144  005046                          CLR     -(SP)
  (13)  025146  153716  002445                  BISB    RLDRV+1,(SP)
  (12)  025152  012746  005423                  MOV     #DRVNAM,-(SP)
  (11)  025156  013746  002440                  MOV     RLBAS,-(SP)
  (10)  025162  012746  005412                  MOV     #BASADD,-(SP)
   (9)  025166  012746  010206                  MOV     #OPR1A,-(SP)
   (8)  025172  012746  007752                  MOV     #OPR6,-(SP)
   (7)  025176  012746  011606                  MOV     #FMTOP1,-(SP)
   (6)  025202  012746  000007                  MOV     #7,-(SP)
   (3)  025206  010600                          MOV     SP,R0
   (4)  025210  104017                          EMT     C$PNTF
   (4)  025212  062706  000020                  ADD     #20,SP
 5493   025216  005037  003656                  CLR     OBUFF           ;CLEAR FOR RESPONSE
 5494   025222                                  GMANIL  #OPR002,OBUFF,1,NO
   (3)  025222  104043                          EMT     C$GMAN
   (3)  025224  000404                          BR      10000$
   (4)  025226  003656                          .WORD   OBUFF
   (5)  025230  000120                          .WORD   T$CODE
   (5)  025232  007514                          .WORD   #OPR002
   (5)  025234  000001                          .WORD   1
   (3)  025236                          10000$:
 5495   025236  005737  003656                  TST     OBUFF           ;TEST RESPONSE YES
 5496   025242  001740                          BEQ     26$             ;NO - SKIP
 5497   025244                          29$:
 5498
 5499   025244                          ENDTST
   (3)  025244                          L10023:
   (3)  025244  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 5          DRIVE SELECT
                                        BGNTST                   ;TEST05
                                                                                T5::
                        012737  000002  002430          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
                        005737  002652                  TST     PASNUM          ;TEST IF FIRST PASS
                        001173                          BNE     EXT05           ;NO - SKIP
                        032737  000004  014446          BIT     #DRSELT,MISWIW  ;TEST IF SELECT TESTS
                        001567                          BEQ     EXT05           ;NO - SKIP
                                        1$:             PRINTF  #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST REMOVE A
                        005046                          CLR     -(SP)
                        153716  002445                  BISB    RLDRV+1,(SP)
                        012746  005423                  MOV     #DRVNAM,-(SP)
                        013746  002440                  MOV     RLBAS,-(SP)
                        012746  005412                  MOV     #BASADD,-(SP)
                        012746  010206                  MOV     #OPR1A,-(SP)
                        012746  010005                  MOV     #OPR7,-(SP)
                        012746  011606                  MOV     #FMTOP1,-(SP)
                        012746  000007                  MOV     #7,-(SP)
                        010600                          MOV     SP,R0
                        104017                          EMT     C$PNTF
                        062706  000020                  ADD     #20,SP
                        005037  003656                  CLR     OBUFF           ;CLEAR FOR RESPONSE
                                                        GMANIL  #OPR002,OBUFF,1,NO
                        104043                          EMT     C$GMAN
                        000404                          BR      10000$
                        003656                          .WORD   OBUFF
                        000120                          .WORD   T$CODE
                        007514                          .WORD   #OPR002
                        000001                          .WORD   1
                                        10000$:
                        005737  003656                  TST     OBUFF           ;TEST RESPONSE YES
                        001740                          BEQ     1$              ;NO - SKIP
                        012737  006152  002424  3$:     MOV     #T0SERR,ERHEAD  ;SET ERROR HEADER MESSAGE
                        004737  016202                  JSR     PC,TSTINT               ;INITIALIZE TEST
                        004737  016234                  JSR     PC,GSTATC       ;DO SELECT AND GET STATUS
                                                        #T504$
                        013737  002444  002530          MOV     RLDRV,TEMPO     ;STORE ORIGINAL DRIVE NUMBER
                        013701  002444                  MOV     RLDRV,R1        ;PUT IT IN R1
                        012704  000004                  MOV     #4,R4           ;SET COUNT FOR NUMBER OF PLUGS
                                        LPT05:  ADD     #400,R1         ;BUMP TO NEXT DRIVE
                        022701  002000                  CMP     #2000,R1        ;CHECK IF TO LARGE
                        001001                          BNE     4$              ;NO - SKIP
                        005001                          CLR     R1              ;ELSE CLEAR TO DRIVE 0
                        010137  002444          4$:     MOV     R1,RLDRV        ;PUT IT BACK IN RLDRV
                                        5$:             PRINTF  #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
                        012746  010222                  MOV     #UNDTST,-(SP)
                        012746  010212                  MOV     #OPR1B,-(SP)
                        005046                          CLR     -(SP)
                        153716  002445                  BISB    RLDRV+1,(SP)
                        012746  010034                  MOV     #OPR8,-(SP)
                        012746  011657                  MOV     #FMTOP3,-(SP)
                        012746  000005                  MOV     #5,-(SP)
                        010600                          MOV     SP,R0
                        104017                          EMT     C$PNTF
                        062706  000014                  ADD     #14,SP
                                                                        ;INSERT PLUG REQUEST
```

```
 5527  025512  005037  003656                CLR     OBUFF           ;CLEAR FOR RESPONSE
 5528  025516                                GMANIL  #OPR002,OBUFF,1,NO
   (3) 025516  104043                        EMT     C$GMAN
   (3) 025520  000404                        BR      10001$
   (4) 025522  003656                        .WORD   OBUFF
   (5) 025524  000120                        .WORD   T$CODE
   (5) 025526  007514                        .WORD   #OPR002
   (5) 025530  000001                        .WORD   1
   (3) 025532                        10001$:
 5529  025532  005737  003656                TST     OBUFF           ;TEST RESPONSE YES
 5530  025536  001744                        BEQ     5$              ;NO - SKIP
 5531  025540                        BGNSUB
   (3) 025540                                                                TS.1:
   (3) 025540  104002                        EMT     C$BSUB
 5532  025542  004737  016234                JSR     PC,GSTATC       ;GET STATUS - REPORT ANY ERROR
 5533  025546  025550                        #60$
 5534  025550  012737  000002  002430  60$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 5535
 5536  025556                        ENDSUB
   (3) 025556                        L10026:
   (3) 025556  104003                        EMT     C$ESUB
 5537  025560  005304                        DEC     R4              ;DEC COUNT
 5538  025562  001322                        BNE     LPTO5           ;LOOP IF NOT ZERO
 5539  025564  013737  002530  002444        MOV     TEMPO,RLDRV     ;ELSE RESTORE RLDRV
 5540  025572                        T504$:
 5541  025572                        4$:     PRINTF  #FMT1,#OPR8,#OPR9
   (9) 025572  012746  010053                MOV     #OPR9,-(SP)
   (8) 025576  012746  010034                MOV     #OPR8,-(SP)
   (7) 025602  012746  011700                MOV     #FMT1,-(SP)
   (6) 025606  012746  000003                MOV     #3,-(SP)
   (3) 025612  010600                        MOV     SP,R0
   (4) 025614  104017                        EMT     C$PNTF
   (4) 025616  062706  000010                ADD     #10,SP
 5542  025622  005037  003656                CLR     OBUFF           ;CLEAR FOR RESPONSE
 5543  025626                                GMANIL  #OPR002,OBUFF,1,NO
   (3) 025626  104043                        EMT     C$GMAN
   (3) 025630  000404                        BR      10000$
   (4) 025632  003656                        .WORD   OBUFF
   (5) 025634  000120                        .WORD   T$CODE
   (5) 025636  007514                        .WORD   #OPR002
   (5) 025640  000001                        .WORD   1
   (3) 025642                        10000$:
 5544  025642  005737  003656                TST     OBUFF           ;TEST RESPONSE YES
 5545  025646  001751                        BEQ     4$              ;NO - SKIP
 5546  025650                        EXTO5:
 5547  025650                        ENDTST
   (3) 025650                        L10025:
   (3) 025650  104001                        EMT     C$ETST
 5548
```

```
5550                                    .SBTTL  *TEST 6          DRIVE SELECT TEST
5551   025652                           BGNTST                   ;TEST06
  (3)  025652                                                                T6::
5552   025652  005737  002652                   TST     PASNUM           ;CHECK IF FIRST PASS
5553   025656  001004                           BNE     1$               ;NO - SKIP
5554   025660  032737  000004  014446           BIT     #DRSELT,MISWIW   ;CHECK IF TEST DRIVE SELECT
5555   025666  001002                           BNE     4$               ;YES - SKIP
5556   025670                           1$:     EXIT    TST
  (3)  025670  104032                           EMT     C$EXIT
  (3)  025672  000630                           .WORD   L10027-.
5557   025674  012737  006106  002424   4$:     MOV     #GSTER1,ERHEAD   ;SET ERROR HEADER
5558   025702  004737  016202                   JSR     PC,TSTINT                ;INITIALIZE TEST
5559   025706  013703  002654                   MOV     PSETNM,R3        ;GET PARAM SET NUMBER
5560   025712  023727  002014  000001           CMP     L$UNIT,#1        ;TEST IF MORE THAN 1 UNIT
5561   025720  101450                           BLOS    5$               ;NO - SKIP
5562   025722  005203                   2$:     INC     R3               ;BUMP PARAMETER SET NUMBER
5563   025724  020337  002014                   CMP     R3,L$UNIT        ;CHECK IF PAST VALID PARAMETER TABLE
5564   025730  101401                           BLOS    3$               ;NO - SKIP
5565   025732  005003                           CLR     R3               ;ELSE CLEAR TO POINT TO ENTRY 0
5566   025734                           3$:     GPHARD  R3,R0
  (3)  025734  010300                           MOV     R3,R0
  (3)  025736  104042                           EMT     C$GPHRD
5567   025740                                   BNCOMPLETE       2$      ;SKIP IF NOT AVAILABLE
  (2)  025740  103370                           BCC     2$
5568   025742  010004                           MOV     R0,R4            ;PUT POINTER INTO R4
5569   025744  021437  002440                   CMP     (R4),RLBAS       ;CHECK IF SAME CONTROLLER
5570   025750  001364                           BNE     2$               ;NO - SKIP
5571   025752  005037  002420                   CLR     DONE             ;CLEAR DONE FLAG
5572   025756  012737  016250  002446           MOV     #GSTAT,L.CS      ;LOAD GET STATUS
5573   025764  056437  000006  002446           BIS     6(R4),L.CS       ;INSERT DRIVE
5574   025772  012737  000013  002452           MOV     #GETSTAT!DRSET,L.DA      ;SET UP TO CLEAR DRIVE
5575   026000  013762  002452  000004           MOV     L.DA,RLDA(R2)    ;LOAD DA REG
5576   026006  013762  002446  000000           MOV     L.CS,RLCS(R2)    ;LOAD CS REG
5577   026014                                   WAITMS  #3               ;WAIT 300 MS
  (3)  026014  012700  000003                   MOV     #3,R0
  (3)  026020  104026                           EMT     C$WTM
5578   026022  005737  002420                   TST     DONE             ;TEST  IF INTERRUPT
5579   026026  001735                           BEQ     2$               ;NO - SKIP
5580   026030  032737  100000  002456           BIT     #ANYERR,T.CS     ;TEST IF ANY ERROR SET
5581   026036  001415                           BEQ     7$               ;NO - GO TEST
5582   026040  000730                           BR      2$               ;ELSE CHECK NEXT DRIVE
5583   026042                           5$:     PRINTF  #FMT9,#OPR10     ;REPORT CAN'T FIND 2ND DRIVE
  (8)  026042  012746  010070                   MOV     #OPR10,-(SP)
  (7)  026046  012746  012117                   MOV     #FMT9,-(SP)
  (6)  026052  012746  000002                   MOV     #2,-(SP)
  (3)  026056  010600                           MOV     SP,R0
  (4)  026060  104017                           EMT     C$PNTF
  (4)  026062  062706  000006                   ADD     #6,SP
5584   026066  000137  026522                   JMP     LCLEXT
5585   026072  016437  000006  002532   7$:     MOV     6(R4),TEMP1      ;STORE NEW ADDRESS
5586                                                                       ;ASK FOR PLUG CHANGE
5587   026100  013700  002444           9$:     MOV     RLDRV,R0         ;GET DRIVE UNDER TEST
5588   026104  013705  002532                   MOV     TEMP1,R5         ;GET NEW ADDRESS
5589   026110  042700  002000                   BIC     #2000,R0         ;CLEAR FOR ADDRESS 0 TO 3
5590   026114  042700  002000                   BIC     #2000,R0
5591   026120  020527  001400           20$:    CMP     R5,#1400         ;TEST IF DRIVE NUMBER 3
```

```
 5592  026124  001001                          BNE     21$                     ;NO - SKIP
 5593  026126  005005                          CLR     R5                      ;ELSE SET TO DRIVE NUMBER 0
 5594  026130  062705  000400          21$:    ADD     #400,R5                 ;BUMP TO NEXT ADDRESS
 5595  026134  020500                          CMP     R5,R0                   ;THIS EQUAL TO NEW ADDRESS?
 5596  026136  001770                          BEQ     20$                     ;YES - SKIP
 5597  026140  052705  000200                  BIS     #CRDYMSK,R5             ;ELSE SET CONTROLLER READY BIT
 5598  026144  010562  000000                  MOV     R5,RLCS(R2)             ;AND LOAD CS REG
 5599  026150                                  PRINTF  #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
   (11)  026150  005046                          CLR     -(SP)
   (11)  026152  153716  002533                  BISB    TEMP1+1,(SP)
   (10)  026156  012746  010212                  MOV     #OPR1B,-(SP)
    (9)  026162  005046                          CLR     -(SP)
    (9)  026164  153716  002445                  BISB    RLDRV+1,(SP)
    (8)  026170  012746  010034                  MOV     #OPR8,-(SP)
    (7)  026174  012746  011635                  MOV     #FMTOP2,-(SP)
    (6)  026200  012746  000005                  MOV     #5,-(SP)
    (3)  026204  010600                          MOV     SP,R0
    (4)  026206  104017                          EMT     C$PNTF
    (4)  026210  062706  000014                  ADD     #14,SP
 5600  026214  005037  003656                  CLR     OBUFF                   ;CLEAR FOR RESPONSE
 5601  026220                                  GMANIL  #OPR002,OBUFF,1,NO
    (3)  026220  104043                          EMT     C$GMAN
    (3)  026222  000404                          BR      10000$
    (4)  026224  003656                          .WORD   OBUFF
    (5)  026226  000120                          .WORD   T$CODE
    (5)  026230  007514                          .WORD   #OPR002
    (5)  026232  000001                          .WORD   1
    (3)  026234                          10000$:
 5602  026234  005737  003656                  TST     OBUFF                   ;TEST IF RESPONSE YES
 5603  026240  001717                          BEQ     9$                      ;NO - SKIP
 5604  026242  012704  000012                  MOV     #10.,R4                 ;SET COUNT
 5605  026246                          BGNSUB                                                  T6.1:
    (3)  026246
    (3)  026246  104002                          EMT     C$BSUB
 5606  026250  013737  002444  002446  8$:     MOV     RLDRV,L.CS              ;SET UP TO SELECT MULTIPLE DRIVES
 5607  026256  013762  002446  000000          MOV     L.CS,RLCSR(R2)          ;DO IT
 5608  026264                                  WAITMS  #10.
    (3)  026264  012700  000012                  MOV     #10.,R0
    (3)  026270  104026                          EMT     C$WTM
 5609  026272  052737  000104  002446          BIS     #GTSTAT,L.CS            ;SET GET STATUS
 5610  026300  012737  000003  002452          MOV     #GETSTAT,L.DA
 5611  026306  013762  002452  000004          MOV     L.DA,RLDA(R2)
 5612  026314  005037  002420                  CLR     DONE
 5613  026320  013762  002446  000000          MOV     L.CS,RLCSR(R2)          ;DO GET STATUS
 5614  026326                                  WAITUS  #1                      ;WAIT FOR INTERRUPT
    (3)  026326  012700  000001                  MOV     #1,R0
    (3)  026332  104027                          EMT     C$WTU
 5615  026334  005737  002420                  TST     DONE                    ;CHECK IF INTERRUPTED
 5616  026340  001012                          BNE     12$                     ;YES - SKIP
 5617  026342  004737  016050                  JSR     PC,WAITIN               ;WAIT FOR TIMEOUT
 5618  026346  012603                          MOV     (SP)+,R3                ;GET ERROR POINTER
 5619  026350  001406                          BEQ     12$                     ;SKIP IF 0
 5620  026352                                  ERRHRD  601.,GSTER1,ERR1
    (3)  026352  104463                          TRAP    T$ERCODE
    (5)  026354  001131                          .WORD   601
    (5)  026356  006106                          .WORD   GSTER1
```

```
   (5)  026360  012630                          .WORD   ERR1
5621    026362                                 EXIT    SUB
   (3)  026362  104032                          EMT     C$EXIT
   (3)  026364  000062                          .WORD   L10030-.
5622    026366                          12$:    WAITMS  #2              ;WAIT FOR DSE TO SET
   (3)  026366  012700  000002                  MOV     #2,R0
   (3)  026372  104026                          EMT     C$WTM
5623    026374  004737  016250                  JSR     PC,GSTAT        ;GET STATUS
5624    026400  026440                          60$
5625    026402  032737  000400  002464          BIT     #DSESTAT,T.MP   ;TEST IF DRIVE SELECT ERROR SET
5626    026410  001007                          BNE     16$             ;YES - SKIP
5627    026412  012703  010732                  MOV     #MDSERR,R3      ;SET NAME MESSAGE POINTER
5628    026416                                  ERRHRD  602,,ERR3
   (3)  026416  104443                          TRAP    T$ERCODE
   (5)  026420  001132                          .WORD   602
   (5)  026422  012744                          .WORD   ERR3
5629    026424                                 EXIT    SUB
   (3)  026424  104032                          EMT     C$EXIT
   (3)  026426  000020                          .WORD   L10030-.
5630    026430  010562  000000          16$:    MOV     R5,RLCS(R2)     ;LOAD IN DIFFERENT ADDRESS
5631    026434  005304                          DEC     R4              ;DEC COUNT
5632    026436  001304                          BNE     8$              ;LOOP IF NOT ZERO
5633    026440  012737  000002  002430  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
5634    026446                                  ENDSUB
   (3)  026446                                  L10030:
   (3)  026446  104003                          EMT     C$ESUB
5635    026450                          15$:    PRINTF  #FMT9,#OPR11    ;REQUEST PLUG CHANGE
   (8)  026450  012746  010136                  MOV     #OPR11,-(SP)
   (7)  026454  012746  012117                  MOV     #FMT9,-(SP)
   (6)  026460  012746  000002                  MOV     #2,-(SP)
   (3)  026464  010600                          MOV     SP,R0
   (4)  026466  104017                          EMT     C$PNTF
   (4)  026470  062706  000006                  ADD     #6,SP
5636    026474  005037  003656                  CLR     OBUFF           ;CLEAR FOR RESPONSE
5637    026500                                  GMANIL  #OPR002,OBUFF,1,NO
   (3)  026500  104043                          EMT     C$GMAN
   (3)  026502  000404                          BR      10000$
   (4)  026504  003656                          .WORD   OBUFF
   (5)  026506  000120                          .WORD   T$CODE
   (5)  026510  007514                          .WORD   #OPR002
   (5)  026512  000001                          .WORD   1
   (3)  026514                          10000$:
5638    026514  005737  003656                  TST     OBUFF           ;TEST RESPONSE YES
5639    026520  001753                          BEQ     15$             ;NO - SKIP
5640    026522                          LCLEXT:
5641    026522                          ENDTST
   (3)  026522                          L10027:
   (3)  026522  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 7          INITIAL STATE
5644  026524                            BGNTST                   ;TEST 07
(3)   026524                                                                    T7::
5645  026524  012737  006137  002424          MOV     #INITST,ERHEAD  ;SET ERROR HEADER
5646  026532  004737  016202                  JSR     PC,TSTINT       ;INITIALIZE TEST
5647  026536                                  WAITUS  #10.            ;WAIT 1 MS
(3)   026536  012700  000012                  MOV     #10.,R0
(3)   026542  104027                          EMT     C$WTU
5648  026544  004737  016234                  JSR     PC,GSTATC       ;GET STATUS
5649  026550  027034                          #65$
5650  026552  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;CHECK IF DRIVE READY
5651  026560  001003                          BNE     3$              ;YES-SKIP
5652  026562  012703  010501                  MOV     #MDRDY,R3       ;SET NAME MESSAGE POINTER
5653  026566  000427                          BR      9$              ;GO REPORT
5654  026570  012703  000005          3$:     MOV     #5,R3           ;SET EXPECTED STATE VALUE
5655
5656  026574  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE OK
5657  026600  001405                          BEQ     5$              ;YES-SKIP
5658  026602                                  ERRHRD  701,,ERR7       ;ELSE REPORT STATE ERROR
(3)   026602  104443                          TRAP    T$ERCODE
(5)   026604  001275                          .WORD   701
(5)   026606  014010                          .WORD   ERR7
5659  026610                                  EXIT    TST             ;EXIT
(3)   026610  104032                          EMT     C$EXIT
(3)   026612  000222                          .WORD   L10031-.
5660  026614  013701  002464          5$:     MOV     T.MP,R1         ;GET MP REG
5661  026620  032701  000020                  BIT     #HOSTAT,R1      ;CHECK HEADS OUT
5662  026624  001003                          BNE     7$              ;YES-SKIP
5663  026626  012703  010721                  MOV     #MHOSTA,R3      ;SET NAME MESSAGE PTR
5664  026632  000405                          BR      9$              ;GO REPORT
5665  026634  032701  000010          7$:     BIT     #BHSTAT,R1      ;CHECK BRUSH HOME SET
5666  026640  001007                          BNE     10$             ;YES-SKIP
5667  026642  012703  010675                  MOV     #MBHSTA,R3      ;SET NAME MESSAGE PTR
5668  026646                          9$:     ERRHRD  702,,ERR3       ;REPORT ERROR
(3)   026646  104443                          TRAP    T$ERCODE
(5)   026650  001276                          .WORD   702
(5)   026652  012744                          .WORD   ERR3
5669  026654                                  EXIT    TST             ;EXIT
(3)   026654  104032                          EMT     C$EXIT
(3)   026656  000156                          .WORD   L10031-.
5670  026660  005737  014446          10$:    TST     MISWIW          ;TEST IF MANUAL INTERVENTION RUN
5671  026664  100034                          BPL     16$             ;NO-SKIP
5672  026666  005737  002652                  TST     PASNUM          ;CHECK IF FIRST PASS
5673  026672  001031                          BNE     16$             ;NO-SKIP
5674  026674  032701  000100                  BIT     #HSSTAT,R1      ;ELSE CHECK HD 0 SELECTED
5675  026700  001411                          BEQ     13$             ;YES-SKIP
5676  026702  012703  010634                  MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
5677  026706  012704  011553                  MOV     #CCYLUP,R4      ;SET CONDITION POINTER
5678  026712                                  ERRHRD  703,,ERR4       ;REPORT ERROR
(3)   026712  104443                          TRAP    T$ERCODE
(5)   026714  001277                          .WORD   703
(5)   026716  013012                          .WORD   ERR4
5679  026720                                  EXIT    TST             ;EXIT
(3)   026720  104032                          EMT     C$EXIT
(3)   026722  000112                          .WORD   L10031-.
5680  026724  032701  001000          13$:    BIT     #VCSTAT,R1      ;CHECK VOL CHECK SET
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-17
DZRLCA.P11     05-OCT-77 10:52          *TEST 7          INITIAL STATE                                    SEQ 0091

 5681   026730  001003                          BNE     15$                     ;YES-SKIP
 5682   026732  012703  010651                  MOV     #MVOLCK,R3              ;ELSE SET NAME MESSAGE PTR
 5683   026736  000743                          BR      9$                      ;GO REPORT
 5684   026740  032737  040000  002456  15$:    BIT     #DRVERR,T.CS            ;TEST DRIVE ERROR SET
 5685   026746  001003                          BNE     16$                     ;YES-SKIP
 5686   026750  012703  010623                  MOV     #MDRERR,R3              ;ELSE SET NAME MESSAGE PTR
 5687   026754  000734                          BR      9$                      ;GO REPORT
 5688   026756  032701  020000          16$:    BIT     #WLSTAT,R1              ;CHECK WRITE LOCK STATUS
 5689   026762  001405                          BEQ     17$                     ;SKIP IF RESET
 5690   026764  012703  010710                  MOV     #MWLSTA,R3              ;ELSE SET NAME MESSAGE PTR
 5691   026770                                  ERRHRD  705,,ERR2
   (3)  026770  104443                          TRAP    T$ERCODE
   (5)  026772  001301                          .WORD   705
   (5)  026774  012676                          .WORD   ERR2
 5692   026776  042701  021177          17$:    BIC     #21177,R1               ;CLEAR STAUS EXCEPT FOR ERROR BITS
 5693   027002  005701                          TST     R1                      ;CHECK IF ANY ERROR SET
 5694   027004  001405                          BEQ     19$                     ;NO-SKIP
 5695   027006                          18$:    ERRHRD  704,,ERR6               ;ELSE REPORT ALL ERRORS
   (3)  027006  104443                          TRAP    T$ERCODE
   (5)  027010  001300                          .WORD   704
   (5)  027012  013132                          .WORD   ERR6
 5696   027014                                  EXIT    TST                     ;EXIT
  (3)   027014  104032                          EMT     C$EXIT
   (3)  027016  000016                          .WORD   L10031-.
 5697   027020  013701  002456          19$:    MOV     T.CS,R1                 ;GET CS REG
 5698   027024  042701  141777                  BIC     #141777,R1              ;CLEAR ALL BUT ERROR BITS
 5699   027030  005701                          TST     R1                      ;TEST IF ANY ERROR SET
 5700   027032  001365                          BNE     18$                     ;YES-SKIP TO REPORT
 5701   027034                          25$:
 5702   027034                          65$:
 5703   027034                          ENDTST
   (3)  027034                          L10031:
   (3)  027034  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 8         INITIAL RESET STATE
027036                                  BGNTST                  ;TEST 8
(3)  027036                                                                     T8::
027036  012737  006137  002424                  MOV     #INITST,ERHEAD
027044  004737  016202                          JSR     PC,TSTINT               ;INITIALIZE TEST

027050  004737  016220                          JSR     PC,GSTATR       ;GET STATUS WITH RESET
027054  027120                                  #65$
027056  005737  014446                          TST     MISWIW          ;CHECK IF MAN INTERVENTION WAS RUN
027062  100016                                  BPL     4$              ;NO-SKIP
027064  005737  002652                          TST     PASNUM          ;CHECK IF 1ST PASS
027070  001013                                  BNE     4$              ;NO-SKIP
027072  032737  000100  002464                  BIT     #HSSTAT,T.MP    ;CHECK HD SELECT STILL 0
027100  001407                                  BEQ     4$              ;YES-SKIP
027102  012703  010634                          MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
027106  012704  011553                          MOV     #CCYLUP,R4      ;SET CONDITION POINTER
027112                                          ERRHRD  801,,ERR4       ;REPORT ERROR
(3)  027112  104443                             TRAP    T$ERCODE
(5)  027114  001441                             .WORD   801
(5)  027116  013012                             .WORD   ERR4
027120                                  4$:
027120                                  65$:
027120                                  ENDTST
(3)  027120                             L10032:
(3)  027120  104001                             EMT     C$ETST
```

```
                                        .SBTTL  *TEST 9          DRIVE READY
                                        BGNTST                   ;TEST 9
                                                                                T9::
                        012737  006165  002424          MOV     #T09ERR,ERHEAD  ;SET ERROR HEADER
                        012701  002514                  MOV     #NEWCYL,R1      ;GET POINTER TO DESIRED LOC
                        005021                          CLR     (R1)+           ;CLEAR NEW CYL
                        005021                          CLR     (R1)+           ;CLEAR CURRENT CYL
                        005021                          CLR     (R1)+           ;       DIFFERENCE
                        005011                          CLR     (R1)            ;       SIGN
                        004737  016202                  JSR     PC,TSTINT               ;INITIALIZE TEST
                        004737  016220                  JSR     PC,GSTATR       ;GET STATUS WITH RESET
                        027420                          #65$
                        004737  020264                  JSR     PC,POSHSB       ;POSITION HEAD SELECTED BIT
                        010537  002524                  MOV     R5,DESHD        ;STORE AS DESIRED HEAD
                        004737  017102                  JSR     PC,SIMSEK       ;EXECUTE SIMPLE SEEK
                        027420                          #65$
                        012703  010501                  MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
                        012704  011512                  MOV     #CDRDY,R4       ;SET CONDITION POINTER
                        004737  016250                  JSR     PC,GSTAT        ;GET STATUS
                        027420                          #65$
                        032737  000001  002456          BIT     #DRDYMSK,T.CS   ;TEST READY SET
                        001405                          BEQ     4$              ;NO-SKIP
                                                        ERRHRD  901,,ERR4       ;REPORT READY ERROR
                        104443                          TRAP    T$ERCODE
                        001605                          .WORD   901
                        013012                          .WORD   ERR4
                                                        EXIT    TST             ;EXIT
                        104032                          EMT     C$EXIT
                        000166                          .WORD   L10033-.
                        012701  000121          4$:     MOV     #81.,R1         ;SET WAIT COUNT
                        004737  016250          5$:     JSR     PC,GSTAT        ;GET STATUS
                        027420                          #65$
                        012703  000005                  MOV     #5,R3           ;SET EXPECTED STATE VALUE
                        023703  002472                  CMP     T.STAT,R3       ;CHECK STATE IS 5
                        001405                          BEQ     7$              ;YES-SKIP
                                                        ERRHRD  902,,ERR7       ;ELSE REPORT
                        104443                          TRAP    T$ERCODE
                        001606                          .WORD   902
                        014010                          .WORD   ERR7
                                                        EXIT    TST
                        104032                          EMT     C$EXIT
                        000130                          .WORD   L10033-.
                        012703  010501          7$:     MOV     #MDRDY,R3
                        032737  000001  002456          BIT     #DRDYMSK,T.CS   ;CHECK READY SET
                        001013                          BNE     12$             ;YES-SKIP
                        005301                          DEC     R1              ;ELSE DEC WAIT COUNT
                        001404                          BEQ     9$              ;SKIP IF 0
                                                        WAITUS  #1
                        012700  000001                  MOV     #1,R0
                        104027                          EMT     C$WTU
                        000747                          BR      5$
                                                9$:     ERRHRD  903,,ERR5       ;REPORT READY ERROR
                        104443                          TRAP    T$ERCODE
                        001607                          .WORD   903
```

```
  (5)  027326  013062                        .WORD   ERR5
5769   027330                               EXIT    TST
  (3)  027330  104032                        EMT     C$EXIT
  (3)  027332  000066                        .WORD   L10033-.
5770
5771   027334  005737  002456        12$:    TST     T.CS            ;TEST IF ANY ERROR
5772   027340  100005                        BPL     15$             ;NO-SKIP
5773   027342                               ERRHRD  904.,,ERR6
  (3)  027342  104443                        TRAP    T$ERCODE
  (5)  027344  001610                        .WORD   904
  (5)  027346  013132                        .WORD   ERR6
5774   027350                               EXIT    TST
  (3)  027350  104032                        EMT     C$EXIT
  (3)  027352  000046                        .WORD   L10033-.
5775   027354  012703  010634        15$:    MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
5776   027360  004737  020264                JSR     PC,POSHSB       ;POSITION HEAD SELECT BIT FOR TEST
5777   027364  020537  002524                CMP     R5,DESHD        ;CHECK IF CORRECT HEAD SELECTED
5778   027370  001413                        BEQ     20$             ;YES-SKIP
5779   027372  005737  002524                TST     DESHD           ;ELSE TEST IF I DESIRED
5780   027376  001405                        BEQ     17$             ;NO-REPORT SB 0
5781   027400                               ERRHRD  905.,,ERR3      ;ELSE REPORT SB 1
  (3)  027400  104443                        TRAP    T$ERCODE
  (5)  027402  001611                        .WORD   905
  (5)  027404  012744                        .WORD   ERR3
5782   027406                               EXIT    TST
  (3)  027406  104032                        EMT     C$EXIT
  (3)  027410  000010                        .WORD   L10033-.
5783   027412                       17$:    ERRHRD  906.,,ERR2
  (3)  027412  104443                        TRAP    T$ERCODE
  (5)  027414  001612                        .WORD   906
  (5)  027416  012676                        .WORD   ERR2
5784   027420                       20$:
5785   027420                       65$:
5786   027420                       ENDTST
  (3)  027420                       L10033:
  (3)  027420  104001                        EMT     C$ETST
```

```
                                        .SBTTL  *TEST 10         SEEK SIGN SWITCH
5789    027422                          BGNTST                   ;TEST 10
  (3)   027422                                                                    T10::
5790    027422  012737  006175  002424        MOV     #T10ERR,ERHEAD  ;SET ERROR HEADER
5791    027430  012701  002514                MOV     #NEWCYL,R1
5792    027434  005021                        CLR     (R1)+           ;CLEAR NEW CYL
5793    027436  005021                        CLR     (R1)+           ;CLEAR CURRENT CYLINDER
5794    027440  005021                        CLR     (R1)+           ;CLEAR DIFFERENCE
5795    027442  052721  000001                BIS     #BIT0,(R1)+     ;SET FOR SIGN OF 1
5796    027446  004737  020264                JSR     PC,POSHSB       ;GET SELECTED HEAD
5797    027452  010521                        MOV     R5,(R1)+        ;SET AS DESIRED HEAD
5798    027454                  T104$:
5799    027454                  BGNSUB
  (3)   027454                                                                    T10.1:
  (3)   027454  104002                        EMT     C$BSUB
5800    027456  004737  016202                JSR     PC,TSTINT               ;INITIALIZE TEST
5801    027462  004737  016220                JSR     PC,GSTATR       ;GET STATUS
5802    027466  027722                        #60$
5803    027470  004737  017102                JSR     PC,SIMSEK       ;DO SEEK
5804    027474  027722                        #60$
5805    027476  012703  010501                MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
5806    027502  012704  011512                MOV     #CDRDY,R4       ;SET CONDITION MESSAGE PTR
5807    027506  004737  016250                JSR     PC,GSTAT        ;GET STATUS
5808    027512  027722                        #60$
5809    027514  032737  000001  002456        BIT     #DRDYMSK,T.CS   ;CHECK READY RESET
5810    027522  001405                        BEQ     4$              ;YES-SKIP
5811    027524                                ERRHRD  1001.,,ERR4     ;REPORT READY ERROR
  (3)   027524  104443                        TRAP    T$ERCODE
  (5)   027526  001751                        .WORD   1001
  (5)   027530  013012                        .WORD   ERR4
5812    027532                                EXIT    SUB             ;EXIT SUBTEST
  (3)   027532  104032                        EMT     C$EXIT
  (3)   027534  000166                        .WORD   L10035-.
5813
5814
5815    027536  012701  000121          4$:   MOV     #81.,R1         ;SET WAIT COUNT
5816    027542  004737  016250          5$:   JSR     PC,GSTAT        ;GET STATUS
5817    027546  027722                        #60$
5818    027550  012703  000005                MOV     #5,R3           ;SET EXPECTED STATE
5819    027554  020337  002472                CMP     R3,T.STAT       ;CHECK STATE IS 5
5820    027560  001405                        BEQ     7$              ;YES-SKIP
5821    027562                                ERRHRD  1002.,,ERR7     ;REPORT STATE ERROR
  (3)   027562  104443                        TRAP    T$ERCODE
  (5)   027564  001752                        .WORD   1002
  (5)   027566  014010                        .WORD   ERR7
5822    027570                                EXIT    SUB             ;EXIT
  (3)   027570  104032                        EMT     C$EXIT
  (3)   027572  000130                        .WORD   L10035-.
5823    027574  012703  010501          7$:   MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
5824    027600  032737  000001  002456        BIT     #DRDYMSK,T.CS   ;CHECK READY SET
5825    027606  001013                        BNE     12$             ;YES-SKIP
5826    027610  005301                        DEC     R1              ;DO WAIT COUNT
5827    027612  001404                        BEQ     9$              ;SKIP IF 0
5828    027614                                WAITUS  #1
  (3)   027614  012700  000001                MOV     #1,R0
  (3)   027620  104027                        EMT     C$WTU
```

```
 5829  027622  000747                          BR      5$
 5830
 5831  027624                          9$:     ERRHRD  1003.,,ERR5     ;REPORT READY ERROR
  (3)  027624  104443                          TRAP    T$ERCODE
  (5)  027626  001753                          .WORD   1003
  (5)  027630  013062                          .WORD   ERR5
 5832  027632                                  EXIT    SUB             ;EXIT
  (3)  027632  104032                          EMT     C$EXIT
  (3)  027634  000066                          .WORD   L10035-.
 5833  027636  005737  002456          12$:    TST     T.CS            ;TEST IF ANY OTHER ERROR
 5834  027642  100005                          BPL     15$             ;NO-SKIP
 5835  027644                                  ERRHRD  1004.,,ERR6     ;REPORT ALL ERRORS
  (3)  027644  104443                          TRAP    T$ERCODE
  (5)  027646  001754                          .WORD   1004
  (5)  027650  013132                          .WORD   ERR6
 5836  027652                                  EXIT    SUB             ;EXIT
  (3)  027652  104032                          EMT     C$EXIT
  (3)  027654  000046                          .WORD   L10035-.
 5837
 5838  027656  012703  010634          15$:    MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
 5839  027662  004737  020264                  JSR     PC,POSHSB       ;GET SELECTED HEAD BIT
 5840  027666  020537  002524                  CMP     R5,DESHD        ;CHECK IF CORRECT
 5841  027672  001413                          BEQ     20$             ;YES - SKIP
 5842  027674  005737  002524                  TST     DESHD           ;WAS IT SET
 5843  027700  001405                          BEQ     17$             ;NO-SKIP
 5844  027702                                  ERRHRD  1005.,,ERR3     ;REPORT SB 1
  (3)  027702  104443                          TRAP    T$ERCODE
  (5)  027704  001755                          .WORD   1005
  (5)  027706  012744                          .WORD   ERR3
 5845  027710                                  EXIT    SUB
  (3)  027710  104032                          EMT     C$EXIT
  (3)  027712  000010                          .WORD   L10035-.
 5846  027714                          17$:    ERRHRD  1006.,,ERR2     ;REPORT SB 0
  (3)  027714  104443                          TRAP    T$ERCODE
  (5)  027716  001756                          .WORD   1006
  (5)  027720  012676                          .WORD   ERR2
 5847
 5848  027722                          20$:
 5849  027722                          60$:
 5850  027722                          ENDSUB
  (3)  027722                          L10035:
  (3)  027722  104003                          EMT     C$ESUB
 5851  027724  005737  002522                  TST     DESSGN          ;CHECK IF BOTH SIGN USED
 5852  027730  001404                          BEQ     25$             ;YES-SKIP
 5853  027732  005037  002522                  CLR     DESSGN          ;SET FOR SIGN OF 0
 5854  027736  000137  027454                  JMP     T104$           ;DO TEST AGAIN
 5855  027742                          25$:
 5856  027742                          ENDTST
  (3)  027742                          L10034:
  (3)  027742  104001                          EMT     C$ETST
```

```
 5858
 5859
 5860                                        .SBTTL  *TEST 11         HEAD ALIGNMENT SUPPORT
 5861   027744                               BGNTST                   ;TEST 11
   (3)  027744                                                                          T11::
 5862   027744  032737  000010  014446              BIT     #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
 5863   027752  001411                            BEQ     1$              ;NO-EXIT
 5864   027754  005737  002652                    TST     PASNUM          ;TEST IF PASS 0
 5865   027760  001006                            BNE     1$              ;NO-EXIT
 5866   027762  023737  002444  002422            CMP     RLDRV,HADONE    ;TEST IF HEAD ALIGN DONE THIS DRIVE
 5867   027770  001004                            BNE     2$              ;NO - SKIP
 5868   027772  000137  030274                    JMP     T115$           ;GO CHECK WRITE LOCK
 5869   027776                           1$:     EXIT    TST
   (3)  027776  104032                            EMT     C$EXIT
   (3)  030000  000374                            .WORD   L10036-.
 5870   030002  013737  002444  002422  2$:     MOV     RLDRV,HADONE    ;SET HEAD ALIGN DONE FLAG
 5871   030010                                   PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11)  030010  005046                            CLR     -(SP)
  (11)  030012  153716  002445                    BISB    RLDRV+1,(SP)
  (10)  030016  012746  005423                    MOV     #DRVNAM,-(SP)
   (9)  030022  013746  002440                    MOV     RLBAS,-(SP)
   (8)  030026  012746  005412                    MOV     #BASADD,-(SP)
   (7)  030032  012746  011733                    MOV     #FMT5,-(SP)
   (6)  030036  012746  000005                    MOV     #5,-(SP)
   (3)  030042  010600                            MOV     SP,R0
   (4)  030044  104017                            EMT     C$PNTF
   (4)  030046  062706  000014                    ADD     #14,SP
 5872   030052                                   PRINTF  #FMT9,#HAMES1   ;TYPE INSTRUCTIONS
   (8)  030052  012746  007357                    MOV     #HAMES1,-(SP)
   (7)  030056  012746  012117                    MOV     #FMT9,-(SP)
   (6)  030062  012746  000002                    MOV     #2,-(SP)
   (3)  030066  010600                            MOV     SP,R0
   (4)  030070  104017                            EMT     C$PNTF
   (4)  030072  062706  000006                    ADD     #6,SP
 5873   030076                                   PRINTF  #FMT9,#HAMES2
   (8)  030076  012746  007442                    MOV     #HAMES2,-(SP)
   (7)  030102  012746  012117                    MOV     #FMT9,-(SP)
   (6)  030106  012746  000002                    MOV     #2,-(SP)
   (3)  030112  010600                            MOV     SP,R0
   (4)  030114  104017                            EMT     C$PNTF
   (4)  030116  062706  000006                    ADD     #6,SP
 5874   030122                           BGNSUB
   (3)  030122                                                                          T11.1:
   (3)  030122  104002                            EMT     C$BSUB
 5875   030124  004737  016202           3$:     JSR     PC,TSTINT               ;INITIALIZE TEST
 5876   030130  005037  002420                    CLR     DONE            ;CLEAR DONE
 5877   030134  013737  002444  002446            MOV     RLDRV,L.CS      ;SET UP FOR GET STATUS
 5878   030142  052737  000104  002446            BIS     #GTSTAT,L.CS
 5879   030150  012737  000013  002452            MOV     #GETSTAT!DRSET,L.DA
 5880   030156  013762  002452  000004            MOV     L.DA,RLDA(R2)
 5881   030164  013762  002446  000000            MOV     L.CS,RLCSR(R2)  ;DO GET STATUS
 5882   030172                                   WAITMS  #50.            ;WAIT FOR INTERRUPT
   (3)  030172  012700  000062                    MOV     #50.,R0
   (3)  030176  104026                            EMT     C$WTM
 5883   030200  005737  002420                    TST     DONE            ;CHECK IF DONE
 5884   030204  001747                            BEQ     3$              ;NO-GO CLR CONTROLLER
```

```
 5885
 5886
 5887  030206  012737  000021  002452  10$:    MOV     #HDSEL!MBSET0,L.DA;LOAD FOR HEAD 1
 5888  030214  032737  020000  002464          BIT     #WLSTAT,T.MP      ;CHECK IF WRITE LOCK SET
 5889  030222  001003                          BNE     12$               ;YES-SKIP
 5890  030224  042737  000020  002452          BIC     #HDSEL,L.DA       ;ELSE CLEAR TO HEAD 0
 5891  030232  013737  002444  002446  12$:    MOV     RLDRV,L.CS        ;LOAD IN DRIVE NUMBER
 5892  030240  052737  000106  002446          BIS     #SEEK,L.CS        ;SET FOR SEEK
 5893  030246  013762  002452  000004          MOV     L.DA,RLDA(R2)     ;LOAD & EXECUTE SEEK
 5894  030254  013762  002446  000000          MOV     L.CS,RLCSR(R2)
 5895  030262                                  WAITMS  #30.              ;WAIT FOR INTERRUPT
   (3) 030262  012700  000036                  MOV     #30.,R0
   (3) 030266  104026                          EMT     C$WTM
 5896  030270  000715                          BR      3$                ;LOOP
 5897  030272                          59$:
 5898  030272                          ENDSUB
   (3) 030272                          L10037:
   (3) 030272  104003                          EMT     C$ESUB
 5899  030274                          T115$:
 5900  030274                          BGNSUB                                                  T11.2:
   (3) 030274
   (3) 030274  104002                          EMT     C$BSUB
 5901  030276  004737  016202                  JSR     PC,TSTINT                 ;INITIALIZE TEST
 5902  030302  004737  016220                  JSR     PC,GSTATR         ;CLEAR DRIVE
 5903  030306  030372                          #60$
 5904  030310  032737  020000  002464          BIT     #WLSTAT,T.MP      ;CHECK WRITE LOCK RESET
 5905  030316  001425                          BEQ     19$               ;YES-SKIP
 5906  030320                          18$:    PRINTF  #FMT9,#OPR12      ;REQUEST WRITE LOCK RESET
   (8) 030320  012746  010167                  MOV     #OPR12,-(SP)
   (7) 030324  012746  012117                  MOV     #FMT9,-(SP)
   (6) 030330  012746  000002                  MOV     #2,-(SP)
   (3) 030334  010600                          MOV     SP,R0
   (4) 030336  104017                          EMT     C$PNTF
   (4) 030340  062706  000006                  ADD     #6,SP
 5907  030344  005037  003656                  CLR     OBUFF                     ;CLEAR FOR RESPONSE
 5908  030350                                  GMANIL  #OPR002,OBUFF,1,NO        ;GET RESPONSE
   (3) 030350  104043                          EMT     C$GMAN
   (3) 030352  000404                          BR      10000$
   (4) 030354  003656                          .WORD   OBUFF
   (5) 030356  000120                          .WORD   T$CODE
   (5) 030360  007514                          .WORD   #OPR002
   (5) 030362  000001                          .WORD   1
   (3) 030364                          10000$:
 5909  030364  005737  003656                  TST     OBUFF             ;WAS ANSWER YES
 5910  030370  001753                          BEQ     18$               ;NO-REPEAT REQUEST
 5911  030372                          19$:
 5912  030372                          60$:
 5913  030372                          ENDSUB
   (3) 030372                          L10040:
   (3) 030372  104003                          EMT     C$ESUB
 5914  030374                          20$:
 5915  030374                          ENDTST
   (3) 030374                          L10036:
   (3) 030374  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 12         HEAD SWITCHING
030376                                  BGNTST                   ;TEST 12
(3)  030376                                                                      T12::
030376  012737  006215  002424                  MOV     #T12ERR,ERHEAD  ;SET ERROR HEADER
030404  012701  002514                          MOV     #NEWCYL,R1      ;GET POINTER TO DESIRED LOCATION
030410  005021                                  CLR     (R1)+           ;CLEAR NEW CYLINDER
030412  005021                                  CLR     (R1)+           ;CLEAR CURRENT CYL.
030414  005021                                  CLR     (R1)+           ;CLEAR DIFFERENCE
030416  005021                                  CLR     (R1)+           ;CLEAR SIGN
030420  012721  000001                          MOV     #1,(R1)+        ;SET FOR HEAD 1
030424                                  T124$:
030424                                  BGNSUB
(3)  030424                                                                      T12.1:
(3)  030424  104002                             EMT     C$BSUB
030426  004737  016202                          JSR     PC,TSTINT               ;INITIALIZE TEST
030432  004737  016220                          JSR     PC,GSTATR       ;GET STATUS WITH RESET
030436  030672                                  #60$
030440  004737  017102                          JSR     PC,SIMSEK       ;DO SEEK
030444  030672                                  #60$
030446  012703  010501                          MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
030452  012704  011512                          MOV     #CDRDY,R4       ;SET CONDITION POINTER
030456  004737  016250                          JSR     PC,GSTAT        ;GET STATUS
030462  030672                                  #60$
030464  032737  000001  002456                  BIT     #DRDYMSK,T.CS   ;CHECK IF READY
030472  001405                                  BEQ     5$              ;NO-SKIP
030474                                          ERRHRD  1201,,ERR4      ;REPORT READY ERROR
(3)  030474  104443                             TRAP    T$ERCODE
(5)  030476  002261                             .WORD   1201
(5)  030500  013012                             .WORD   ERR4
030502                                  EXIT    SUB                     ;EXIT
(3)  030502  104032                             EMT     C$EXIT
(3)  030504  000166                             .WORD   L10042-.

030506  012701  000121          5$:     MOV     #81.,R1         ;SET WAIT COUNT
030512  004737  016250          6$:     JSR     PC,GSTAT        ;GET STATUS
030516  030672                          #60$
030520  012703  000005                  MOV     #5,R3           ;SET EXPECTED STATE VALUE
030524  020337  002472                  CMP     R3,T.STAT       ;CHECK IF STATE IS 5
030530  001405                          BEQ     7$              ;YES-SKIP
030532                                  ERRHRD  1202,,ERR7      ;REPORT STATE ERROR
(3)  030532  104443                     TRAP    T$ERCODE
(5)  030534  002262                     .WORD   1202
(5)  030536  014010                     .WORD   ERR7
030540                                  EXIT    SUB
(3)  030540  104032                     EMT     C$EXIT
(3)  030542  000130                     .WORD   L10042-.

030544  012703  010501          7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
030550  032737  000001  002456          BIT     #DRDYMSK,T.CS   ;CHECK DRIVE READY
030556  001013                          BNE     12$             ;YES-SKIP
030560  005301                          DEC     R1              ;DEC WAIT COUNT
030562  001404                          BEQ     9$              ;SKIP IF 0
030564                                  WAITUS  #1
(3)  030564  012700  000001             MOV     #1,R0
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-26
DZRLCA.P11    05-OCT-77 10:52          *TEST 12        HEAD SWITCHING                                  SEQ 0100

   (3)  030570  104027                          EMT     C$WTU
 5959   030572  000747                          BR      6$
 5960
 5961   030574                          9$:     ERRHRD  1203.,,ERR5      ;REPORT READY ERROR
   (3)  030574  104443                          TRAP    T$ERCODE
   (5)  030576  002263                          .WORD   1203
   (5)  030600  013062                          .WORD   ERR5
 5962   030602                                  EXIT    SUB             ;EXIT
   (3)  030602  104032                          EMT     C$EXIT
   (3)  030604  000066                          .WORD   L10042-.
 5963
 5964   030606  005737  002456          12$:    TST     T.CS            ;TEST IF ANY ERROR
 5965   030612  100005                          BPL     15$             ;NO-SKIP
 5966   030614                                  ERRHRD  1204.,,ERR6     ;REPORT ALL ERRORS
   (3)  030614  104443                          TRAP    T$ERCODE
   (5)  030616  002264                          .WORD   1204
   (5)  030620  013132                          .WORD   ERR6
 5967   030622                                  EXIT    SUB
   (3)  030622  104032                          EMT     C$EXIT
   (3)  030624  000046                          .WORD   L10042-.
 5968   030626  012703  010634          15$:    MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
 5969   030632  004737  020264                  JSR     PC,POSHSB       ;POSITION HEAD SELECT BIT
 5970   030636  023705  002524                  CMP     DESHD,R5        ;CHECK IF CORRECT HEAD SELECTED
 5971   030642  001413                          BEQ     20$             ;YES-SKIP
 5972   030644  005737  002524                  TST     DESHD           ;WAS HEAD 0 SELECTED
 5973   030650  001405                          BEQ     17$             ;YES-SKIP
 5974   030652                                  ERRHRD  1205.,,ERR3     ;REPORT HEAD SB 1
   (3)  030652  104443                          TRAP    T$ERCODE
   (5)  030654  002265                          .WORD   1205
   (5)  030656  012744                          .WORD   ERR3
 5975   030660                                  EXIT    SUB             ;EXIT
   (3)  030660  104032                          EMT     C$EXIT
   (3)  030662  000010                          .WORD   L10042-.
 5976   030664                          17$:    ERRHRD  1206.,,ERR2     ;ELSE REPORT HEAD SB 0
   (3)  030664  104443                          TRAP    T$ERCODE
   (5)  030666  002266                          .WORD   1206
   (5)  030670  012676                          .WORD   ERR2
 5977
 5978   030672                          20$:
 5979   030672                          60$:
 5980   030672                          ENDSUB
   (3)  030672                          L10042:
   (3)  030672  104003                          EMT     C$ESUB
 5981   030674  005737  002524                  TST     DESHD           ;CHECK IF HD 0 WAS DONE
 5982   030700  001404                          BEQ     25$             ;YES-SKIP
 5983   030702  005037  002524                  CLR     DESHD           ;ELSE SET TO HEAD 0
 5984   030706  000137  030424                  JMP     T124$           ;REDO TEST
 5985   030712                          25$:
 5986   030712                          ENDTST
   (3)  030712                          L10041:
   (3)  030712  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 13         READ HEADER (PART 1)
030714                                  BGNTST                   ;TEST 13
(3)  030714                                                                        T13::
030714  012737  006227  002424                  MOV     #T13ERR,ERHEAD  ;SET ERROR HEADER
030722  012701  002514                          MOV     #NEWCYL,R1      ;GET ADDRESS OF DESIRED LOCATIONS
030726  005021                                  CLR     (R1)+           ;CLEAR NEW CYL
030730  005021                                  CLR     (R1)+           ;CLEAR CURRENT CYL
030732  005021                                  CLR     (R1)+           ;CLEAR DIFF
030734  005021                                  CLR     (R1)+           ;CLEAR SIGN
030736  005021                                  CLR     (R1)+           ;CLEAR HEAD
030740                                  T134$:
030740                                  BGNSUB
(3)  030740                                                                        T13.1:
(3)  030740  104002                             EMT     C$BSUB
030742  004737  016202                          JSR     PC,TSTINT               ;INITIALIZE TEST
030746  004737  016220                          JSR     PC,GSTATR       ;GET STATUS W/RESET
030752  031040                                  #60$
030754  004737  017102                          JSR     PC,SIMSEK       ;DO SEEK
030760  031040                                  #60$
030762  012701  000121                          MOV     #81.,R1         ;SET WAIT COUNT
030766  004737  020314                          JSR     PC,RDYWAIT      ;WAIT FOR READY
030772  031040                                  #60$

030774  004737  017626                  10$:    JSR     PC,XRDHDC       ;DO READ HEADER
031000  031040                                  #60$
031002  012703  010634                          MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
031006  004737  020256                          JSR     PC,POSHW1       ;POSITION HS BIT IN HD WRD 1
031012  020537  002524                          CMP     R5,DESHD        ;CHECK IF HEAD CORRECT
031016  001410                                  BEQ     15$             ;YES-SKIP
031020                                          ERRHRD  1301.,ERR3      ;REPORT SB 1
(3)  031020  104443                             TRAP    T$ERCODE
(5)  031022  002425                             .WORD   1301
(5)  031024  012744                             .WORD   ERR3
031026                                          EXIT    SUB
(3)  031026  104032                             EMT     C$EXIT
(3)  031030  000010                             .WORD   L10044-.
031032                                  17$:    ERRHRD  1302.,ERR2      ;REPORT SB 0
(3)  031032  104443                             TRAP    T$ERCODE
(5)  031034  002426                             .WORD   1302
(5)  031036  012676                             .WORD   ERR2

031040                                  15$:
031040                                  60$:
031040                                  ENDSUB
031040                                  L10044:
(3)  031040  104003                             EMT     C$ESUB
031042  005737  002524                          TST     DESHD           ;TEST IF HEAD 1 DONE
031046  001007                                  BNE     20$             ;YES-SKIP
031050  012737  000001  002524                  MOV     #1,DESHD        ;ELSE SET TO HEAD 1
031056  013737  002464  002530                  MOV     HDWRD1,TEMPO    ;STORE HDR WORD 1
031064  000725                                  BR      T134$           ;DO TEST AGAIN
031066  042737  100177  002530  20$:    BIC     #↑CHDCYL,TEMPO  ;CLEAR ALL BUT CYLINDER IN 1ST HEADER
031074  042737  100177  002464          BIC     #↑CHDCYL,HDWRD1 ;CLEAR ALL BY CYL IN 2ND HEADER
031102  023737  002530  002464          CMP     TEMPO,HDWRD1    ;COMPARE IF EQUAL
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-28
DZRLCA.P11    05-OCT-77 10:52        *TEST 13        READ HEADER (PART 1)                    SEQ 0102

 6031  031110  001405                        BEQ     22$                ;YES-SKIP
 6032  031112  012703  007273                MOV     #CYLPER,R3         ;SET NAME MESSAGE PTR
 6033  031116                                ERRHRD  1306.,,ERR1        ;REPORT HEAD ALIGNMENT PROBLEM
   (3) 031116  104443                        TRAP    T$ERCODE
   (5) 031120  002432                        .WORD   1306
   (5) 031122  012630                        .WORD   ERR1
 6034  031124                          22$:
 6035  031124                          ENDTST
   (3) 031124                          L10043:
   (3) 031124  104001                        EMT     C$ETST
```

```
                                .SBTTL  *TEST 14         READ HEADER (PART 2)
031126                          BGNTST                   ;TEST 14
031126                                                                  T14::
031126  012737  006243  002424          MOV     #T14ERR,ERHEAD  ;SET ERROR HEADER
031134  012701  002516                  MOV     #CURCYL,R1      ;GET ADDRESS OF DESIRED VALUE
031140  005021                          CLR     (R1)+           ;CLEAR CURRENT CYL
031142  005021                          CLR     (R1)+           ;CLEAR DESIRED DIFF
031144  005021                          CLR     (R1)+           ;CLEAR SIGN
031146  005021                          CLR     (R1)+           ;CLEAR DESIRED HEAD
031150                          T153$:
031150                          BGNSUB
031150                                                                  T14.1:
031150  104002                          EMT     C$BSUB
031152  004737  016202                  JSR     PC,TSTINT               ;INITIALIZE TEST
031156  004737  016220                  JSR     PC,GSTATR       ;CLEAR DRIVE
031162  031356                          #60$
031164  004737  017102                  JSR     PC,SIMSEK       ;DO SEEK
031170  031356                          #60$
031172  012701  000310                  MOV     #200.,R1        ;SET WAIT COUNT FOR 20 MS
031176  004737  020314                  JSR     PC,RDYWAIT      ;WAIT FOR READY
031202  031356                          #60$
031204  004737  020710                  JSR     PC,RDALHD       ;DO READ HEADER ALL HEADERS
031210  031356                          #60$
031212  005037  002426                  CLR     MORECE          ;CLEAR MORE COMPARE ERRORS FOR REPORT
031216  052737  000002  002416          BIS     #HDRCMP,OPFLAG  ;SET HDR COMPARE FLAG
031224  005003                          CLR     R3              ;CLR FOR HDR COUNT
031226  012704  003256                  MOV     #IBUFF,R4       ;GET POINTER FOR HDR TO BE CHECKED
031232  012705  002530                  MOV     #TEMP0,R5       ;GET POINTER TO TEST AREA
031236  012701  000050                  MOV     #40.,R1         ;SET HDR COUNT
031242  011415                          MOV     (R4),(R5)       ;GET FIRST HEADER WORD
031244  042715  100100                  BIC     #BIT15!HDHSEL,(R5);CLEAR BIT 15 AND HEAD SEL BIT
031250  005737  002524                  TST     DESHD           ;TEST IF HD 0 DESIRED
031254  001404                          BEQ     10$             ;YES-SKIP
031256  052715  000100                  BIS     #HDHSEL,(R5)    ;ELSE SET HEAD BIT
031262  005065  000002                  CLR     2(R5)           ;CLEAR 2ND WORD OF TEST AREA
031266  021524                  10$:    CMP     (R5),(R4)+      ;COMPARE HEADER WORD
031270  001405                          BEQ     13$             ;SKIP IF OK
031272  005744                          TST     -(R4)           ;ELSE POSITION R4 TO BAD WORD
031274                                  ERRHRD  1501.,,ERR10    ;REPORT ERROR
104443                                  TRAP    T$ERCODE
002735                                  .WORD   1501
014220                                  .WORD   ERR10
031302  005724                          TST     (R4)+           ;BUMP R4 TO NEXT WORD
031304  005203                  13$:    INC     R3              ;BUMP WORD COUNT
031306  005724                          TST     (R4)+           ;TEST 2ND WORD IS 0
031310  001405                          BEQ     15$             ;YES - SKIP
031312  022544                          CMP     (R5)+,-(R4)     ;POSITION PTRS FOR REPORT
031314                                  ERRHRD  1501.,,ERR10    ;REPORT ERROR
104443                                  TRAP    T$ERCODE
002735                                  .WORD   1501
014220                                  .WORD   ERR10
031322  024524                          CMP     -(R5),(R4)+     ;REPOSITION POINTER
031324  005724                  15$:    TST     (R4)+           ;POSITION R4 PAST ECC WORD
031326  005203                          INC     R3              ;BUMP WORD COUNT
031330  005215                          INC     (R5)            ;BUMP SECTOR COUNT
031332  011500                          MOV     (R5),R0         ;CHECK IF SECTOR IS PAST LAST SECTOR
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 84-30
DZRLCA.P11    05-OCT-77 10:52          #TEST 14        READ HEADER (PART 2)                    SEQ 0104

 6084   031334  042700  177700                  BIC     #^CHDSEC,R0
 6085   031340  022700  000050                  CMP     #40.,R0
 6086   031344  001002                          BNE     17$                     ;NO-SKIP
 6087   031346  042715  000077                  BIC     #HDSEC,(R5)             ;ELSE CLEAR SECTOR TO 0
 6088   031352  005301                  17$:    DEC     R1                      ;DEC HDR COUNT
 6089   031354  001344                          BNE     10$                     ;YES-SKIP
 6090
 6091   031356                          60$:
 6092   031356                          ENDSUB
   (3)  031356                          L10046:
   (3)  031356  104003                          EMT     C$ESUB
 6093   031360  005737  002524                  TST     DESHD                   ;CHECK IF HD 1 TESTED
 6094   031364  001005                          BNE     20$                     ;YES-SKIP
 6095   031366  012737  000001  002524          MOV     #1,DESHD                ;ELSE SET TO HEAD 1
 6096   031374  000137  031150                  JMP     T153$                   ;REDO TEST
 6097   031400                          20$:
 6098   031400                          ENDTST
   (3)  031400                          L10045:
   (3)  031400  104001                          EMT     C$ETST
 6099   031402                          ENDMOD
```

```
                                        BGNMOD  HRDPRM
                                        BGNHRD
  (3)  031402  000025                           .WORD L10047-L$HARD/2
                                        GPRML   CNTYPE,CNT,1,YES
  (4)  031404  004130                           .WORD   T$CODE
  (4)  031406  031520                           .WORD   CNTYPE
  (4)  031410  000001                           .WORD   1
                                        GPRMA   CSRMSG,CSR,0,160000,177776,YES
  (4)  031412  000031                           .WORD   T$CODE
  (4)  031414  031456                           .WORD   CSRMSG
  (4)  031416  160000                           .WORD   T$LOLIM
  (4)  031420  177776                           .WORD   T$HILIM
                                        GPRMA   VECMSG,VECT,0,0,776,YES
  (4)  031422  001031                           .WORD   T$CODE
  (4)  031424  031472                           .WORD   VECMSG
  (4)  031426  000000                           .WORD   T$LOLIM
  (4)  031430  000776                           .WORD   T$HILIM
                                        GPRMD   BRMSG,PRIOR,0,340,0,7,YES
  (4)  031432  002032                           .WORD   T$CODE
  (4)  031434  031501                           .WORD   BRMSG
  (4)  031436  000340                           .WORD   340
  (4)  031440  000000                           .WORD   T$LOLIM
  (4)  031442  000007                           .WORD   T$HILIM
                                        GPRMD   DRMSG,DRSB,0,3400,0,7,YES
  (4)  031444  003032                           .WORD   T$CODE
  (4)  031446  031512                           .WORD   DRMSG
  (4)  031450  003400                           .WORD   3400
  (4)  031452  000000                           .WORD   T$LOLIM
  (4)  031454  000007                           .WORD   T$HILIM

                                        ENDHRD
  (2)                                           .EVEN
  (3)  031456                           L10047:

  031456  052502  020123  042101  CSRMSG: .ASCIZ  /BUS ADDRESS/
  031464  051104  051505  000123
  031472  042526  052103  051117  VECMSG: .ASCIZ  /VECTOR/
  031500      000
  031501      102  020122  042514  BRMSG:  .ASCIZ  /BR LEVEL/
  031506  042526  000114
  031512  051104  053111  000105  DRMSG:  .ASCIZ  /DRIVE/
  031520  046122  030461      000  CNTYPE: .ASCIZ  /RL11/
  031525                        ENDMOD
          031526                        .EVEN


  031526                        BGNMOD  SFTPRM
  031526                        BGNSFT
  (3)  031526  000021                           .WORD L10050-L$SOFT/2

  031530                        GPRML   SELQ,MISWI,4,YES
  (4)  031530  000130                           .WORD   T$CODE
  (4)  031532  031572                           .WORD   SELQ
  (4)  031534  000004                           .WORD   4
  031536                        GPRML   ALGNQ,MISWI,10,YES
  (4)  031536  000130                           .WORD   T$CODE
  (4)  031540  031625                           .WORD   ALGNQ
```

```
  (4)  031542  000010                        .WORD   10
6131   031544                        GPRML   MANQ,MISWI,100000,YES
  (4)  031544  000130                        .WORD   T$CODE
  (4)  031546  031664                        .WORD   MANQ
  (4)  031550  100000                        .WORD   100000
6132
6144   031552                        3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
  (4)  031552  004052                        .WORD   T$CODE
  (4)  031554  031726                        .WORD   ERLIMQ
  (4)  031556  000377                        .WORD   377
  (4)  031560  000000                        .WORD   T$LOLIM
  (4)  031562  000377                        .WORD   T$HILIM
6148   031564                        GPRML   AUTOQ,MISWI,20,YES
  (4)  031564  000130                        .WORD   T$CODE
  (4)  031566  031752                        .WORD   AUTOQ
  (4)  031570  000020                        .WORD   20
6149   031572                        ENDSFT
  (2)                                       .EVEN
  (3)  031572                        L10050:
6150
6156   031572  054105  041505  052125  SELQ:   .ASCIZ  /EXECUTE DRIVE SELECT TESTS/
       031600  020105  051104  053111
       031606  020105  042523  042514
       031614  052103  052040  051505
       031622  051524  000040
6157   031625  042530  052503  ALGNQ:  .ASCIZ  /EXECUTE HEAD ALIGNMENT SUPPORT/
       031632  042524  044040  040505
       031640  020104  046101  043511
       031646  046516  047105  020124
       031654  052523  050120  051117
       031662  000124
6159   031664  054105  041505  052125  MANQ:   .ASCIZ  /EXECUTE MANUAL INTERVENTION TESTS/
       031672  020105  040515  052516
       031700  046101  044440  052116
       031706  051105  042526  052116
       031714  047511  020116  042524
       031722  052123  000123
6167   031726  050123  041505  043111  ERLIMQ: .ASCIZ  /SPECIFY ERROR LIMIT/
       031734  020131  051105  047522
       031742  020122  044514  044515
       031750  000124
6168   031752  051104  050117  042040  AUTOQ:  .ASCIZ  /DROP DRIVE IF NO RESPONSE/
       031760  044522  042526  044440
       031766  020106  047516  051040
       031774  051505  047520  051516
       032002  000105
6172                                        .EVEN
6173   032004                        ENDMOD
6174   032004                        LASTAD
  (2)                                       .EVEN
```

```
 6175                                   .TITLE  PDP-11 DIAGNOSTIC SUPERVISOR
17471            062704                END.SUPV=.+2
17473            071776                .=71776
17474  071776   000000                 .WORD   0
17476            072000                X1X1=.
17477            000200                .END 200
```

```
ABOFLA  032256 G
ABOPAS  032170 G
ABO.FM  035132
AFMID   002622
AFMIDU  002624
ALGNQ   031625
ALLCYL= 000001
ALLOC   052722
ALLSEC= 000002
ANYERR= 100000
ARMID   002626
ARMIDU  002630
AUTOQ   031752
AUTOSZ= 000020
A$AAW   037042
A$AAX   037056
A$AAY   037064
A$AAZ   037100
A$ABA   037110
BADADD= 004000
BAMSK = 000060
BANAM   005530
BASADD  005412
BELL    011433
BGN.SU= 032004
BHSTAT= 000010
BINMSG  051232
BIT0  = 000001 G
BIT00 = 000001 G
BIT01 = 000002 G
BIT02 = 000004 G
BIT03 = 000010 G
BIT04 = 000020 G
BIT05 = 000040 G
BIT06 = 000100 G
BIT07 = 000200 G
BIT08 = 000400 G
BIT09 = 001000 G
BIT1  = 000002 G
BIT10 = 002000 G
BIT11 = 004000 G
BIT12 = 010000 G
BIT13 = 020000 G
BIT14 = 040000 G
BIT15 = 100000 G
BIT2  = 000004 G
BIT3  = 000010 G
BIT4  = 000020 G
BIT5  = 000040 G
BIT6  = 000100 G
BIT7  = 000200 G
BIT8  = 000400 G
BIT9  = 001000 G
BLD.HW  037722
BLOCK   055056
BRMSG   031501
BSFLAG  002432
BSFVAL  002664
BSNSTR  010400
BYPSNM  010311
B$AAB   041324
B$AAF   041236
CAFDT   011564
CALLPC= 000022
CALLPS= 000024
CALLSP= 000026
CALLTC= 000030
CAL.CL  057440
CAL.TI  057476 G
CAMSK = 077600
CCYLUP  011553
CDRDY   011512
CHKLUP  041340
CHKSTR  053264
CHKTTY  051352
CHK.FO  033550
CHK.MA  037500
CHK.PC  044566
CHK.SW  033272
CHRCNT  052604
CH.FLA  037202
CH.PAS  037224
CKDATA= 000102
CKERLM  015656
CLEAR.  040622
CLKACC  032166 G
CLKBFR  057442
CLKCNT  032164 G
CLKRES  061040 G
CLKSER  061340 G
CLKSON  032230 G
CLK.SE  037302
CLNCOD  015466 G
CLRPAR  022502
CLR.MA  037556
CNT   = 000010
CNTYPE  031520
CNVT    055516
COMMTA  055336
COMPOP= 007777
CONHNG= 000004
CONTCL  061120 G
CONTIN  014666
COSTAT= 000040
COUNT   002646
CRDYMS= 000200
CRLF    051434
CSNAM   005523
CSR   = 000000
CSRMSG  031456
CURCYL  002516
CURR.T  032204 G
CYLPER  007273
CYLTBL  002342
CYLUP = 000004
CYLWD   010276
C$AAD   044540
C$AAE   044552
C$AAK   045370
C$AAL   045500
C$ABRT= 000021
C$ADR = 000020
C$AU  = 000054
C$BRK = 000022
C$BSEG= 000004
C$BSUB= 000002
C$BUFF= 000030
C$CEFG= 000046
C$CLEA= 000012
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000053
C$DRPT= 000024
C$DU  = 000055
C$EDIT= 000006
C$ERDF= 000002
C$ERHP= 000003
C$ERSF= 000001
C$ERSO= 000004
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GMAN= 000043
C$GPHR= 000042
C$GPRI= 000040
C$GTIM= 000052
C$INIT= 000011
C$INLP= 000020
C$KWOF= 000035
C$KWON= 000034
C$LOOP= 000100
C$MANI= 000051
C$MSG = 000023
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$POIN= 000040
C$QIO = 000377
C$RDBU= 000007
C$REFG= 000050
C$REQT= 000045
C$RESE= 000033
C$REVI= 000001
C$RPT = 000025
C$SEFG= 000047
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
C$UNBU= 000031
C$WTM = 000026
C$WTU = 000027
C10MS   011532
C55EC   011576
C500MS  011545
DANAM   005535
DATACM= 000001
DCKERR= 004000
DCLIM = 000012
DCLIMW  014460
DECMSG  051246
DESDIF  002520
DESHD   002524
DESSEC  002526
DESSGN  002522
DIAG.T  032264 G
DIFAUG  002510
DIFWD   010252
DIRBIT= 000004
DIRMSK= 077600
DLTERR= 010000
DONE    002420
DPDVD   062050 G
DPMUL   061736 G
DRDYMS= 000001
DRMSG   031512
DRSB  = 000006
DRSELT= 000004
DRSET = 000010
DRVCNT  002506
DRVERR= 040000
DRVNAM  005423
DRVNAV  005430
DSESTA= 000400
DSMSK = 001400
DSPCOD  014462 G
DUNIT.  032174 G
DVC.FT  045340
D$AAG   046210
D$AAH   046226
D$AAI   051000
D$AAJ   051004
D$AAK   051022
D$AAL   051040
D$AAM   051050
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
EF01  = 000001 G
EF02  = 000002 G
EF03  = 000003 G
EF04  = 000004 G
EF05  = 000005 G
EF06  = 000006 G
EF07  = 000007 G
EF08  = 000010 G
EF09  = 000011 G
EF10  = 000012 G
EF11  = 000013 G
EF12  = 000014 G
EF13  = 000015 G
EF14  = 000016 G
EF15  = 000017 G
EF16  = 000020 G
EMT.TR  032262 G
END.OF  040610
END.SU= 062704
EOP.CH  061362 G
EOP.FM  035146
EOP.IN  037220
ERHEAD  002424
ERLIM = 000010
ERLIMQ  031726
ERLIMW  014456
ERRCNT  002650
ERRFOR  045556
ERRHAN  044572
ERRSWI  002430
ERRVEC  002642
ERR.HR  045350
ERR.SF  045354
ERR1    012630 G
ERRIFQ  045642
ERR10   014220 G
ERR2    012676 G
ERR3    012744 G
ERR4    013012 G
ERR5    013062 G
ERR6    013132 G
ERR7    014010 G
```

```
ERR8    014060 G    F$AU   = 000015    G$RADT= 000100    INIT.R  032020 G    LUP     057344
ERR9    014154 G    F$BGN  = 000040    G$XFER= 000004    INOUTS= 000020      LUP.AD  044570
ESC.PC  044564      F$CLEA= 000007     G$YES = 000010    INPUTA  052210      L$AUT   002070 G
EXACYL  002636      F$DU   = 000016    HADONE  002422    INTEBL= 000100      L$CCP   002044 G
EXHCYL  002634      F$END  = 000041    HAMES1  007357    INTFOR  045506      L$CLEA  015466 G
EXOCYL  002632      F$HARD= 000004     HAMES2  007442    INTHLR  015620 G    L$DEPO  002011 G
EXROT   002640      F$HW   = 000013    HCESTA= 040000    INVAL.  036732      L$DEVP  002052 G
EXTO5   025650      F$INIT= 000006     HCORED  036764    INVINT  045400      L$DISP  014464 G
FBSFIL  003062      F$JMP  = 000050    HCOREQ  036644    INV.SW  033226      L$DR    002102 G
FILL    052102      F$MOD  = 000000    HCORET  032216 G  IN.SUF  040574      L$DRCT  002062 G
FILL.C  000204 G    F$MSG  = 000011    HCRCER= 004000    I$AU   = 000041     L$DRS   002064 G
FLAGS   032226 G    F$PWR  = 000017    HDALIG= 000010    I$CLN  = 000041     L$DRST  002102 G
FLAGTA  055254      F$RPT  = 000012    HDCYL  = 077600   I$DU   = 000041     L$DTP   002040 G
FLAG.I  037266      F$SEG  = 000003    HDHSEL= 000100    I$HRD  = 000041     L$DU    015606 G
FLA.SE  055222      F$SOFT= 000005     HDMOVF  007234    I$INIT= 000041      L$DUT   002072 G
FLG.MA  037226      F$SRV  = 000010    HDRCMP= 000002    I$MOD  = 000041     L$DVTY  002104 G
FMTOP1  011606      F$SUB  = 000002    HDR40  = 100000   I$MSG  = 000041     L$EF    002024 G
FMTOP2  011635      F$SW   = 000014    HDSEC  = 000077   I$PWR  = 000041     L$EXP1  002032 G
FMTOP3  011657      F$TEST= 000001     HDSEL  = 000020   I$RPT  = 000041     L$EXP2  002034 G
FMT1    011700      GARBAG  052606     HDWD    010265    I$SEG  = 000041     L$EXP3  002036 G
FMT1.1  011705      GETCHR  051312     HDWRD1  002464    I$SFT  = 000041     L$HARD  031404 G
FMT11   012124      GETCMN  054676     HDWRD2  002466    I$SRV  = 000041     L$HPCP  002046 G
FMT12   012132      GETPAR  046370     HDWRD3  002470    I$SUB  = 000041     L$HPTP  002056 G
FMT13   012140      GETPOS  020562     HEAD   = 000006    I$TST  = 000041     L$HW    014432 G
FMT14   012204      GETSTA= 000003     HEADLM= 010000    J$JMP  = 000167     L$ICP   002042 G
FMT15   012236      GETSWI  053672     HEADW   014454    KBPTR   032036 G    L$INIT  014520 G
FMT16   012272      GET.TW  053442     HERTZ.  036604    KBUF    032040 G    L$LADP  002076 G
FMT17   012303      GLBDAT  002112 G   HFIN    002602    LABACF  007204      L$LAST  032004 G
FMT18   012325      GLBEQA  002112 G   HFINU   002604    LABACR  007220      L$MREV  002012 G
FMT19   012357      GLBERR  012630 G   HFOUT   002606    LABEXP  007113      L$NAME  002000 G
FMT2    011714      GLBSUB  015612 G   HFOUTU  002610    LABHCF  007154      L$REPP  002054 G
FMT20   012414      GLBTXT  004534 G   HICYL  = 020000    LABHCR  007170      L$REV   002010 G
FMT21   012444      GSTAT   016250     HILIM  = 000004    LABIN   007070      L$SOFT  031530 G
FMT22   012467      GSTATC  016234     HILIMW  014452    LABMID  007076      L$SFC   002030 G
FMT23   012523      GSTATG  016260     HNFERR= 010000    LABOCF  007124      L$SPCP  002050 G
FMT24   012537      GSTATR  016220     HOLDSP= 000020    LABOCR  007140      L$SPTP  002060 G
FMT25   012544      GSTER1  006106     HOSTAT= 000020    LABOUT  007105      L$STA   002066 G
FMT26   012554      GTSTAT= 000104     HPTCOD  014430 G  LAB1    005547      L$SW    014446 G
FMT27   012600      G$EXCP= 000400     HRDPRM  031402 G  LAB2    005562      L$TIML  002022 G
FMT28   012617      G$HILI= 000002     HRDWTS  022532 G  LCLEXT  026522      L$TIMU  002020 G
FMT3    011717      G$LOLI= 000001     HRIN    002612    LINE.F  032260 G    L$TIM1  002016 G
FMT4    011722      G$NO   = 000000    HRINU   002614    LOAD.F  037222      L$TSTI  002074 G
FMT5    011733      G$OFFS= 000400     HROUT   002616    LOCERR  002656      L$UNIT  002014 G
FMT6    011753      G$OFSI= 000376     HROUTU  002620    LOCYL  = 040000     L.BA    002450
FMT7    012015      G$PRMA= 000001     HSMSK  = 000100   LOGMSG  051254      L.CLK.  036570
FMT8    012065      G$PRMD= 000002     HSSTAT= 000100    LOLIM  = 000002     L.CS    002446
FMT9    012117      G$PRML= 000000     H$AAB   056044    LOLIMW  014450      L.DA    002452
FOLWRT= 000100      G$RADA= 000140     IBUFF   003256    LPBFR   032034 G    L.MP    002454
FORM.T  045652      G$RADB= 000000     ININIT  032206 G  LPCNTR  032032 G    L10000  012674
FREE    053160      G$RADD= 000040     INITCO  014520 G  LPT.AD  036622      L10001  012742
FRMWD   010303      G$RADF= 000200     INITIA  051262    LPT.RE  036616      L10002  013010
FWDSKO= 002000      G$RADL= 000120     INITST  006137    LPT05   025430      L10003  013060
FWDSKS= 000400      G$RADO= 000020     INIT.M  037624    LSI.RE  036612      L10004  013130
```

```
L10005  014006          MCYLUP  004774          MSTERR  011015          OPR1A   010206          PRIO4 = 000200 G
L10006  014056          MDATCP  004636          MTMBS   005350          OPR1B   010212          PRIO5 = 000240 GG
L10007  014152          MDCRC   010534          MTOSLO  005603          OPR10   010070          PRIO6 = 000300 GG
L10010  014216          MDHEDR  002000 G        MUL     061604 G        OPR11   010136          PRIO7 = 000340 G
L10011  014426          MDLT    010561          MULOAD  004763          OPR12   010167          PRNTST  052452
L10012  014444          MDRDY   010501          MUNDEF  011235          OPR2    007642          PRO.CM  037174
L10013  014462          MDRERR  010623          MVOLCK  010651          OPR3    007674          PSETNM  002654
L10014  015464          MDRRES  005622          MWDERR  011051          OPR5    007710          PTAB.S  032214 G
L10015  015604          MDRVST  010747          MWGERR  011000          OPR6    007752          PUTCHR  051266
L10016  015610          MDSERR  010732          MWLSTA  010710          OPR7    010005          PWCON   015012
L10017  015654          MEM.SI  036632          MWORD   005575          OPR8    010034          PWRFLG  002662
L10020  023000          MERRS   011424          MWRCHK  004572          OPR9    010053          PWR.FA  062542 GG
L10021  023202          MEXERS  011361          MWRITE  004606          ORIN    002566          PWR.FL  032016 GG
L10022  024436          MFLERR  011122          MWRSET  004717          ORINU   002570          PWR.MS  062670
L10023  025244          MFMTER  005323          MWRTAB  011343          ORMID   002572          PWR.SA  062664
L10024  025142          MFOLWR  005050          M40HDR  004703          ORMIDU  002574          PWR.UP  062666
L10025  025650          MFWDSK  005127          NEWCYL  002514          OROUT   002576          P.CLK.  036576
L10026  025556          MFWSKO  005162          NEWPRI  061330 G        OROUTU  002600          P2T01E  006267
L10027  026522          MGTSTA  004622          NEXTAR  055440          OUTINS= 000040          P2T02E  006267
L10030  026446          MHCERR  011033          NOCLR = 000010          O$APTS= 000000          P2T03E  006306
L10031  027034          MHCRC   010524          NOERCT  002657          O$AU  = 000000          P2T04E  006332
L10032  027120          MHDERR  011105          NOIRPT= 000002          O$BGNR= 000000          P2T05E  006354
L10033  027420          MHDRCP  004661          NOOP  = 000100          O$BGNS= 000001          P2T06E  006376
L10034  027742          MHFCRC  010573          NOPWR   005463          O$DU  = 000001          P2T07E  006420
L10035  027722          MHNF    010545          NO.CLK  034342          O$GNSW= 000001          P2T08E  006444
L10036  030374          MHOSTA  010721          NO.FLA  055234          O$POIN= 000001          P2T09E  006466
L10037  030272          MHSTA   010634          NO.LPT  052552          PARSES  054750          P2T10E  006473
L10040  030372          MINOUT  005025          NO.PTA  037032          PART1 = 000000          P2T11E  006510
L10041  030712          MIN.IN  032004 G        NR    = 000000          PAR.LA  050742          P2T12E  006525
L10042  030672          MIN.US  032006 G        NSTACH  006040          PASCNT  002644          P2T13E  006541
L10043  031124          MISTST  006015          NUMBIN  045676          PASNEW  014714          P2T14E  006561
L10044  031040          MISWI = 000000          NUM.LA  046044          PASNUM  002652          P2T15E  006604
L10045  031400          MISWIW  014446          NUM.UN  032402          PATTBL  002224          P2T16E  006634
L10046  031356          MITEST= 100000          NUNITS  041312          PAT1    004256          P2T17E  006657
L10047  031456          MNDRST  011175          NXMERR= 020000          PAT10   004532          P2T18E  006717
L10050  031572          MNEERR  011150          NXTFOR  055510          PAT2    004260          P2T19E  006744
MAJ.IN  032010 G        MNOCLR  005731          NXTPAS  014706          PAT3    004320          RDALHD  020710
MAJ.LO  057444          MNOINT  005652          OBUFF   003656          PAT4    004360          RDDATA= 000114
MAJ.US  032012 G        MODR    061650 G        OCTMSG  051240          PAT5    004420          RDHEAD= 000110
MANQ    031664          MOPER   004733          OFIN    002552          PAT6    004426          RDNOHR= 000116
MAN.TI  034312          MOPERR  011066          OFINU   002554          PAT7    004466          RDYCHK  017366
MAPROX  007060          MORECE  002426          OFMID   002556          PAT8    004470          RDYWAI  020314
MAP16   062306 G        MOUTIN  005004          OFMIDU  002560          PAT9    004530          READRL  016016
MASK.B  041336          MPNAM   005542          OFOUT   002562          POSHDO  020270          READ.P  057446 GG
MASK.W  041334          MQUALS= 003760          OFOUTU  002564          POSHSB  020264          REGBAC  062272 GG
MBADAD  005252          MREAD   004542          OLDCYL  002512          POSHW1  020256          REGSAV  062256 G
MBADSF  005273          MREADH  004555          OPFLAG  002416          PRINTC  052562          RELDWT= 040000
MBHSTA  010675          MRESKO  005216          OPIERR= 002000          PRINTF  056064          REQN.P  037204
MBSETO= 000001          MREVSK  005074          OPMSGS  002112          PRIOR = 000004          REQN.T  037176
MCERR   010512          MRLFAL  011302          OPR002  007514          PRIO0 = 000000 G        RESE3   011437
MCONHN  005705          MRSLT   004747          OPR003  007541          PRIO1 = 000040 GG       RESE4   011443
MCOSTA  010662          MSEEK   004534          OPR004  010235          PRIO2 = 000100 GG       RESE5   011450
MCYLOC  011170          MSPERR  010762          OPR1    007564          PRIO3 = 000140 G        RESE6   011455
```

```
RESPAR  002474          STARTC  061114 G        TOSLOW= 000001          T.BA    002460          USER.P  032210 G
RESTAR  014656          STATE2  011462          TRPFLG  002660          T.CS    002456          USER.T  032212 G
RESTBL  002164          STATE3  011472          TRPHAN  015612          T.DA    002462          VALDES  007021
REVSKO= 001000          STATE5  011502          TSTINT  016202          T.MP    002464          VALID.  032452
REVSKS= 000200          STOSTA= 010000          TSTLAB  006007          T.STAT  002472          VAL.LA  033202
RE.SET  033374          STRCHR  052142          TST.AB  041450          T05ERR  006152          VAL.SW  037240
RLBA  = 000002          STREQ.  036744          TST.TO  033254          T09ERR  006165          VCNRST  005751
RLBAS   002440          STRT.T  037200          TYPEC   051600          T1      022532 G        VCSTAT= 001000
RLCS  = 000000          ST.REQ  037120          TYPEPC  045474          T10     027422 G        VECMSG  031472
RLCSR = 000000          ST.SET  033444          TYPFLA  055116          T10ERR  006175          VECT  = 000002
RLDA  = 000004          SUBSTK  002250          TYPLIN  051476          T10.1   027454          WAITIN  016050
RLDRV   002444          SUNIT.  037206          TYPNUM  051064          T10$    027454          WCMSK = 017777
RLMP  = 000006          SUPERV  035164          TYPSTR  051516          T11     027744 G        WCRNG = 160000
RLVEC   002442          SUPFLA  032172 G        TYP.ER  045360          T11.1   030122          WDESTA= 100000
RORWOP= 020000          SUPV.T  032350 G        TY.UNI  040614          T11.2   030274          WGESTA= 002000
RPTOP   021252          SUP.PR  033216          T$ARGC= 000002          T11$    030274          WIDTH   046244
RPTREM  022246          SVCBGL= 000001          T$CODE= 000130          T12     030376 G        WLSTAT= 020000
RPTRES  022040          SVCCNT= 177777          T$ERCO= 000043          T12ERR  006215          WRTSWI  002434
RSTACK  061532 G        SVCGBL= 177777          T$ERRN= 002735          T12.1   030424          WTDATA= 000112
RSTRT   014602          SVCHAN  041516          T$EXCP= 000000          T12$    030424          XEQDIA  061416 G
RSX.FL  037216          SVCINS= 000000          T$FLAG= 000040          T13     030714 G        XEQSUB  061404 G
SAMSK = 000077          SVCSTK= 177777          T$HILI= 000377          T13ERR  006227          XEQ.CL  041254
SBSFIL  002666          SVCSUB= 000001          T$LOLI= 000000          T13.1   030740          XEQ.CM  036562
SEARCH  053410          SVCTAG= 000000          T$LSYM= 010000          T13$    030740          XEQ.IN  040736
SECWD   010271          SVCTST= 000001          T$MCAL= 177777          T14     031126 G        XEQ.LA  035120
SEEK  = 000106          SWCHAN  037024          T$NEST= 177777          T14ERR  006243          XEQ.OP  041030
SEEKOP= 010000          SWITCH  055414          T$NSKO= 000000          T14.1   031150          XEQ.PR  034352
SEGSTA  032232 G        SW.PTA  037010          T$NSK1= 000005          T15$    031150          XEQ.TE  041074
SELQ    031572          SYS.FT  045330          T$NSK2= 000002          T16ERR  006257          XRDHD   017636
SEQMES  010324          S$LSYM= 010000          T$SAVL= 177777          T2      023002 G        XRDHDC  017626
SETDON  014734          TBLSTR  002436          T$SEGL= 177777          T25TBL  002274          XRDHDG  017642
SET.MA  037412          TCERR   010456          T$SUBN= 000001          T3      023204 G        XTIME   060124 G
SFTPRM  031526 G        TEMPO   002530          T$TAGL= 177777          T33TBL  002322          XTIMEN  060750
SGNWD   010260          TEMP1   002532          T$TAGN= 010051          T365$   024436          XTIMST  060146
SHIFT   062370 G        TEMP2   002534          T$TEMP= 000000          T4      024440 G        XXDP.D  036772
SIMSEK  017102          TEMP3   002536          T$TEST= 000016          T4.1    024460          X$ALWA= 000000
SIZE.C  061246 G        TEMP4   002540          T$TSTM= 177777          T465$   025134          X$FALS= 000040
SIZE.M  061164 G        TEMP5   002542          T$TSTS= 000001          T5      025246 G        X$OFFS= 000400
SIZ.TR  061324          TEMP6   002544          T$$CLE= 010015          T5.1    025540          X$TRUE= 000020
SKTMES  006756          TEMP7   002546          T$$DU = 010016          T504$   025572          X1X1  = 072000
SPDERR  006052          TEMP8   002550          T$$HAR= 010047          T6      025652 G        $BREG   037300
SPDSTA= 004000          TERMI   057434          T$$HW = 010012          T6.1    026246          $ENDAD  061370 G
SPEC.U  037122          TERMLI  055242          T$$INI= 010014          T7      026524 G        $SAV2   062434 G
SPTCOD  014444 G        TERMTA  051224          T$$MSG= 010011          T8      027036 G        $SAV3   062450 G
SPV.SE  033624          TEST.M  037134          T$$SOF= 010050          T9      027122 G        $SAV4   062466 G
SRTMES  006772          TIMFLG  032162 G        T$$SRV= 010017          ULOAD = 000010          $SAV5   062506 G
SSINDX  002414          TIM.CO  032014 G        T$$SUB= 010046          UNDTST  010222          .     = 072000
STAMES  010367          TIM.OP  045650          T$$SW = 010013          UNI.MA  037124
STAMSK= 000007          TOO.MA  051204          T$$TES= 010045          UNXERR  005772

. ABS.  072000      000

 ERRORS DETECTED:  0

 DSKZ:DZRLCA,DSKZ:DZRLCA/EQ:PART1=DZRLCA.SML,DZRLCA.PT1,DZRLCA.P11,DZRLCA.PT2,DZRLCA.SUP
```

```
OUTERR  MACY11 30(1046)  04-NOV-77  13:14  PAGE 87-4                    SEQ 0112
DZRLCA.SUP    11-OCT-77 15:40          SYMBOL TABLE

 RUN-TIME:  45.49 1 SECONDS
 RUN-TIME RATIO: 1257/96=13.0
 CORE USED:  18K  (35 PAGES)
```