RL11,RLV11 RL01/02 DRIVE TEST 3 AH-F845A-MC
CZRLNA0 FICHE 1 OF 1

IDENTIFICATION

PRODUCT CODE: AC-F843A-MC

PRODUCT NAME: CZRLNA0 RL01/02 DRIVE TEST 3

DATE CREATED: 5-JAN-79

REVISED: 7-DEC-79

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHORS: D. DEKNIS, C. CAMPBELL

TABLE OF CONTENTS

1.0 GENERAL INFORMATION
1.1 PROGRAM ABSTRACT
1.1.1 STRUCTURE OF PROGRAM
1.1.2 DIAGNOSTIC INFORMATION
1.2 SYSTEM REQUIREMENTS
1.2.1 HARDWARE REQUIREMENTS
1.2.2 SOFTWARE REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 HOW TO RUN THIS DIAGNOSTIC
2.1.1 THE FIVE STEPS OF EXECUTION
2.1.2 SAMPLE RUN-THROUGH
2.2 CHAIN MODE OPERATION
2.3 DETAILS OF COMMANDS AND SYNTAX
2.3.1 TABLE OF COMMAND VALIDITY
2.3.2 COMMAND SYNTAX
2.4 EXTENDED P-TABLE DIALOGUE
2.5 HARDWARE PARAMETERS
2.6 SOFTWARE PARAMETERS

3.0 ERROR INFORMATION
3.1 ERROR REPORTING
3.1.2 SPECIFIC RESULT MESSAGES
3.1.3 OTHER MESSAGES
3.2 ERROR HALTS

4.0 PERFORMANCE AND PROGRESS REPORTS
4.1 PERFORMANCE REPORTS
4.2 PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC 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 2.2 ''CHAIN MODE OPERATION'' FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES.

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 (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 ''OPERATING INSTRUCTIONS''.

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 6.25K OF THE UPPER PART OF MEMORY JUST BELOW THE XXDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

1.1.2 DIAGNOSTIC INFORMATION

THIS PROGRAM TESTS AND EXERCISES RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM FIRST TESTS THE RL01/02 SEEK TIMING. DATA TRANSFERS ARE DONE AFTER THE SEEK TIMING TEST. THE FIRST DATA TRANSFER IS READING OF THE BAD SECTOR FILES WHICH ARE STORED AND USED LATER TO PREVENT TESTING ON BAD SECTORS. FOLLOWING DATA READ AND WRITE TESTING, THE PROGRAM TESTS FOR OVERWRITE PROBLEMS AND ADJACENT CYLINDER INTERFERENCE.

THE WRITE LOCK DATA PROTECTION TEST IS PERFORMED IF MANUAL INTERVENTION IS REQUESTED.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

* PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
* CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
* 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH·
  - 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
  - 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
* KW11-P CLOCK (REQUIRED TO PERFORM TESTS 1 AND 4)
* LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLJB0 RL01/02 DRIVE TEST PART 2 (FORMERLY CZRLDB0)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 DISK SUBSYSTEM USER'S GUIDE (EK-RL01-UG-002)
XXDP+/SUPERVISOR USER'S MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAB0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLGB0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLHB0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLIB0	RL01/02 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 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 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

THIS DIAGNOSTIC PROGRAM SHOULD BE LOADED AND STARTED USING NORMAL XXDP+ PROCEDURES. START THE EXECUTION OF THE XXDP+ MONITOR BY USING THE APPROPRIATE BOOTSTRAP PROGRAM. THE MONITOR WILL PRINT A MESSAGE IDENTIFYING ITSELF AND REQUESTING THAT THE CURRENT DATE BE ENTERED. AN EXAMPLE OF THIS MESSAGE IS GIVEN BELOW FOR THE XXDP+ MONITOR:

```
CHMDKA0 XXDP+ DK MONITOR NNK
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YY):
```

AFTER THE DATE HAS BEEN ACCEPTED BY THE MONITOR THE RESTART ADDRESS OF THE MONITOR IS PRINTED. THEN THE FOLLOWING TWO QUESTIONS ARE ASKED:

```
50 HZ ? N
LSI ?  N
```

THE DEFAULTS ARE BOTH "NO". TYPE "R" AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

```
**********
* STEP 1 *
**********
```

THE DIAGNOSTIC WILL ISSUE THE PROMPT "DR>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP+, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP+. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP+ COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE XXDP+ "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP+ DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN 2.3 "DETAILS OF COMMANDS AND SYNTAX". HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS.HOE
```

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE 'DR>'' LEVEL NEED TO BE TYPED.

2. THE ''PASS'' SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.

3. THE ''FLAGS'' SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

PNT	PRINT NUMBER OF TEST BEING EXECUTED
LOE	LOOP ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

   THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

****
* STEP 2 *
****

WHEN YOU HAVE TYPED IN A ''START'' COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION ''# UNITS?'' TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE ''HEADER'' STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS ''HEADER'' STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

****
* STEP 3 *
****

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE ''HARDWARE QUESTIONS''. THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED ''HARDWARE P-TABLES''. ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

****
* STEP 4 *
****

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR ALL THE UNITS, YOU WILL BE ASKED "CHANGE SW?" IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE 'Y'. IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE 'N'. IF YOU TYPE 'Y' YOU WILL BE ASKED THE SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

****
* STEP 5 *
****

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DR>).

2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

   HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

   LOE SET: THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.

   NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURRED.

2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE RE-ISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN)

2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)

3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURRED. NO QUESTIONS ASKED.

4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT ON ERROR).

THE MOST TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY:

```
PRO/FLAGS:IER:LOE:HOE=0
```

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG
2. TURN ON THE LOE FLAG
3. TURN OFF THE HOE FLAG
4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER.

WHEN YOU'VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE.

1. START
2. RESTART
3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START
2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS
(O=OPERATOR, D-DIAGNOSTIC):

```
                                        B`
                                        WHOM
                                        ENTERED:
                                        -------

.R CZRLNA                               O
DRS LOADED                              D
DIAG. RUN-TIME SERVICES REV. D APR-79   D
CZRLN-A-0                               D
CZRLN TESTS SEEK AND ROTATIONAL         D
TIMING & WRITE & READ DATA
UNIT IS RL01, RL02                      D
DR>STA/PASS:1/FLAGS:HOE                 D,O

# UNITS (D) ? 2                         D,O

UNIT 0                                  D
RL11 (L) Y ?                            D,O
BUS ADDRESS (O) 174400 ?                D,O
VECTOR (O) 160 ?                        D,O
DRIVE (O) 0 ?                           D,O
DRIVE TYPE = RL01 (L) Y ?               D,O
BR LEVEL (O) 5 ?                        D,O

UNIT 1                                  D
RL11 (L) Y ?                            D,O
BUS ADDRESS (O) 174400 ?                D,O
VECTOR (O) 160 ?                        D,O
DRIVE (O) 0 ? 1                         D,O
DRIVE TYPE = RL01 (L) ? N               D,O (N-RL02)
BR LEVEL (O) 5 ?                        D,O

CHANGE SW (L) ? Y                       D,O

USE ALL CYL (L) N ?                     D,O
USE ALL SECT (L) N ?                    D,O
DO MANUAL INTERVENTION TEST (L) N ?     D,O
LOW SEEK LIMIT (L) N ?                  D,O
UPPER SEEK LIMIT (L) N ?                D,O
USE ONLY ONE SURF (L) N ?               D,O
INPUT ERROR LIMIT (D) 20 ?              D,O
DATA CMP ERR LMT (D) 10 ?               D,O


CZRLN HRD ERR 00004 TST 003 SUB 002 PC:004130
ERR HLT

DR>PRO/FLAGS:IER:LOE:HOE=0              D,O
```

```
**************************************************
AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
ERROR WITHOUT PRINTING ANYTHING.  YOU CAN SCOPE
THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT
**************************************************

^C                                          O

DR>CON/FLAGS:HOE:IER:LOE=0                  D,O

CHANGE SW (L) ? N                           D,O

CZRLN EOP 1                                 D
^C

DR>RESTART/PASS:1                           D,O

CHANGE SW (L) ? N                           D,O
-------------------
-------------------
-------------------
-------------------
```

2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION.

TO RUN CHAIN MODE, THE XXDP+ MONITOR USES AN ASCII FILE (KNOWN AS A CHAIN FILE) LISTING THE PROGRAMS TO BE RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN. THIS FILE MUST BE ON THE SYSTEM DEVICE.

A CHAIN FILE MAY BE GENERATED BY USE OF THE XTECO TEXT EDITOR. THIS FILE MUST HAVE A CCC EXTENSION. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED BY THE XXDP+ MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENCOUNTERED.

TO EXECUTE A CHAIN FILE THE USER TYPES:

```
C FILNAM <CR> OR
C FILNAM/QV  <CR>
```

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE XXDP+ MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PROGRAM COUNT IS NOT USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OF QUICK VERIFY.

WHEN PROGRAMS ARE RUN IN CHAIN MODE, THE SOFTWARE SWITCH REGISTER SHOULD BE SET TO 000000. THE XXDP+ MONITOR PRINTS EACH COMMAND TAKEN FROM THE CHAIN FILE AND THEN EXECUTES THE COMMAND. WHEN THE LAST COMMAND OTHER THAN ANOTHER C COMMAND HAS BEEN EXECUTED THE XXDP+ MONITOR TERMINATES CHAIN MODE AND TYPES A PROMPT (.). IT IS READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE. IF THE LAST COMMAND IS ANOTHER C COMMAND, THE CHAIN MODE WILL CONTINUE AND THE CHAIN FILE SPECIFIED BY THIS NEW C COMMAND WILL BE USED.

IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERMINATION HE MAY DO SO BY TYPING A CONTROL/C. HOWEVER, THE MONITOR WILL NOT ABORT THE CHAIN MODE UNTIL IT RECEIVES PROGRAM CONTROL FROM THE PROGRAM CURRENTLY RUNNING.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

	HOW ENTERED	LEGAL COMMANDS
1.	OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2.	DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3.	OPERATOR INTERRUPTED THE	START PRINT DISPLAY FLAGS ZFLAGS EXIT

4.	AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

2.3.2 COMMAND SYNTAX

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

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

"TEST-LIST" IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE 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 THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

"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 TEST EXECUTION. "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 MOD , BYPASSING MANUAL INTERVENT'ON TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

ADR EXECUTE AUTODROP CODE

LOT LOOP ON TEST

EVL EVALUATE

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.

''EOP-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.

****
RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS:UNIT-LIST
****

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW ''P-TABLES'' ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.

THE QUESTION ''CHANGE SW?'' IS ASKED AND THE ANSWERS GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. 'UNIT-LIST'' IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO ''ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND''. THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO ''ALL'') OR THE NEXT RESTART.

2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

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

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 RE-EXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART

2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
EXIT
****

RETURN TO XXDP+ PROMPT MODE.

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

THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED BACK OR UNTIL A START COMMAND IS GIVEN. A DROP CANNOT BE FOLLOWED BY A PROCEED.

THERE IS ALSO A 'DROP'' MACRO INTERNAL TO THE DIAGNOSTIC, WHICH GIVES THE FACILITY OF AUTO-DROPPING. THE DURATION OF A PROGRAM DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.

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

THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN PREVIOUSLY DROPPED BY THE DROP COMMAND) TO THE TEST SEQUENCE. AN ADD CANNOT BE FOLLOWED BY A PROCEED.

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

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

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

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.

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

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

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

ALL FLAGS ARE CLEARED.

2.4 EXTENDED 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), 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.

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 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT "BR LEVEL" 5. THE FIRST 4 DRIVES ARE RL01'S AND THE LAST 4 DRIVES ARE RL02'S (ON THE SECOND CONTROLLER):

```
# UNITS (D) ? 8

UNIT 0
RL11 (L) Y ?
BUS ADDDRESS (O) 174400 ?
VECTOR (O) 160 ?
DRIVE (O) 0 ? 0-3
DRIVE TYPE - RL01 (L) Y ?
BR LEVEL (O) 5 ?

UNIT 4
RL11 (L) Y ?
BUS ADDRESS (O) 174400 ? 175400
VECTOR (O) 160 ? 164
DRIVE (O) 0 ? 0-3
DRIVE TYPE = RL01 (L) Y ? N
BR LEVEL (O) 5 ?
```

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE 'BR LEVEL'' (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO 'RL11'' TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE 'BR LEVEL'' FROM THE FIRST PASS.

2.5 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 RESPONSE.

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) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE - RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

CHANGE S.W. ?

A YES ANSWER WILL ASK THE FOLLOWING SOFTWARE PARAMETER QUESTIONS, WITH THE PRESENT DEFAULT VALUE PRINTED TO THE LEFT OF THE QUESTION MARK. (THE LAST ANSWER GIVEN IS THE DEFAULT) THE DEFAULT IS TAKEN ON A <CR>. CONTROL Z (^Z) WILL DEFAULT ALL REMAINING QUESTIONS AND START THE TEST.

USE ALL CYLINDERS (N)?

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

USE ALL SECTORS (N)?

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

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

LOWER SEEK LIMIT (N)?

IF 'YES', THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

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

UPPER SEEK LIMIT (N)?

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

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF 'YES', THE NEXT PARAMETER IS REQUESTED.

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

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

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

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVENT DATA.

3.1 ERROR REPORTING

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

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

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

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

A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

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

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

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

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

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

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

THE QUALIFIERS ' OL 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	(BRUSH 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	(HDR CRC OR DATA CRC ERROP BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HDR 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 TOO LATE''

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

'FAIL TO RELOAD HEADS AFTER ERR CLEAR''

THIS 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''

THIS IS REPORTED WHEN THE PROGRAM CANNOT DETERMINE THE DRIVE STATE OR STATUS.

'WRITE ABORTED''

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

'COULD NOT RETRIEVE DRIVE STATUS''

THIS 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''

THIS 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''

THIS IS REPORTED WHEN THE COMMAND SUCCESSFULLY COMPLETES BUT THE CONTROLLER HAS NOT GENERATED AN INTERRUPT.

'ERR DID NOT CLEAR''

THIS 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''

THIS IS REPORTED WHEN THE GET STATUS W/RESET COMMAND DOES NOT CLEAR ALL DRIVE ERRORS.

'UNEXPECTED ERR''

THIS IS REPORTED WHEN THE CONTROLLER SENSES AN ERROR BUT NO ERROR BITS ARE SET.

'BAD SEC FILE FMT ERR''

THIS 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''

THIS IS REPORTED (WITH THE UNIT NUMBER) WHEN MORE THAN THE SPECIFIED NUMBER OF ERRORS (DEFAULT 20) HAVE OCCURED IN ANY SINGLE PASS.

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

LINE 9 PROVIDES THE CONTENTS OF THE CONTROLLER REGISTERS WHEN THE ERROR BEING REPORTED WAS DETECTED. FREQUENTLY THE REGISTER CONTENTS OF OP INIT AND OP DONE WILL BE DIFFERENT. OP INIT MAY INDICATE A SEEK WAS BEING PERFORMED BUT OP DONE MAY INDICATE THE ERROR WAS DETECTED BY A READ HEADER. THE REASON IS THAT A SEEK WAS EXECUTED AND DID NOT PROPERLY POSITION HEADS AND WHEN THE READ HEADER WAS DONE THE HEADS WERE ON THE WRONG CYLINDER.

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

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

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR
BIT 14 - DRIVE ERROR
BIT 13 - NON EXISTENT 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 15-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (1-40.)

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

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 - DRIVE TYPE IS RL02 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
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 SEEK TIMING

(P-CLOCK IS REQUIRED TO PERFORM THIS TEST.)

POSITION HEADS AT CYLINDER 0.

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

REPEAT ABOVE SEEKING BETWEEN CYLINDER 127 TO 128 AND 254 TO 255 FOR RL01 AND 255 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 TO 127 AND 128 TO 256 FOR RL01 AND CYLINDER 0 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 AND 255 FOR RL01 AND 0 TO 511 FOR RL02.

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

	INNER	MIDDLE	OUTER	MAX TIME
1 CYL FWD	X	X	X	X
1 CYL REV	X	X	X	X
MID CYL FWD	X		X	X
MID CYL REV	X		X	X
MAX CYL FWD		X		X
MAX CYL REV		X		X

THE X INDICATES WHERE TIME WILL BE REPORTED.

TEST 2 BASIC READ DATA TEST
******

POSITION HEADS AT MAX CYLINDER.

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

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

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

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

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

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

TEST 3 WRITE/READ DATA TEST (PART 1)
******

POSITION HEADS AT CYLINDER 0

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

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

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

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

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

TEST 4 ROTATIONAL TIMING TEST
******

(P-CLOCK IS REQUIRED TO PERFORM THIS TEST.)

POSITION HEADS TO CYLINDER 0.

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

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

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

TEST 5 WRITE/READ TEST (PART 2)
******

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

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

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

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

READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

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

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

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

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

TEST 6 WRITE LOCK ERROR AND DATA PROTECTION TEST
******

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

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

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

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

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

TEST 7 ADJACENT CYLINDER INTERFERENCE TEST
******

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

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

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

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

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

TEST 8 OVERWRITE TEST
******

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

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

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

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

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

18	MACRO DEFINITIONS
88	GLOBAL DATA SECTION
222	GLOBAL DATA SECTION
643	GLOBAL MESSAGES
876	ERROR MESSAGES
1213	INITIALIZATION SECTION
1348	AUTO DROP SECTION
1388	CLEANUP CODE SECTION
1418	GLOBAL SUBROUTINES
2671	*TEST 1 **SEEK TIMING
2862	*TEST 2 **BASIC READ DATA (BAD SECTOR FILE)
2956	*TEST 3 **WRITE/READ DATA (PART 1)
3004	*TEST 4 **ROTATIONAL TIMING
3086	*TEST 5 **WRITE/READ DATA (PART 2)
3240	*TEST 6 **WRITE LOCK ERROR AND DATA PROTECTION
3352	*TEST 7 **ADJACENT CYLINDER INTERFERENCE
3523	*TEST 8 **OVERWRITE
3690	PARAMETER CODING

```
                        PART2==1
                                .ENABLE ABS
                                .ENABLE AMA
                        .=2000
                                .MCALL  SVC

002000                          SVC
                        SVCTST=1
                        SVCSUB=1
                        SVCBGL=1
                        SVCINS=0
                        SVCTAG=0
```

```
                        .TITLE  MACRO DEFINITIONS

                        .MACRO  WAITUS  ARG             ;MACRO MICRO-SEC WAIT
                                MOV     ARG,XDELAY      ;SAVE ARGUMENT
                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                        .ENDM

                        .MACRO  WAITMS  ARG             ;MACRO MILLI-SEC WAIT
                                MOV     ARG,YDELAY      ;SAVE ARGUMENT
                                JSR     PC,XTIME        ;CALL TIMING ROUTINE
                        .ENDM

                        .MACRO  ABORTWAIT               ;MACRO CLEAR UNELAPSED TIME
                                MOV     XDELAY,TEMPO    ;SAVE MICRO-SEC RUN TIME
                                MOV     YDELAY,TEMP     ;SAVE MILLI-SEC RUN TIME
                                CLR     XDELAY          ;ABORT MICRO-SEC WAIT
                                CLR     YDELAY          ;ABORT MILLI-SEC WAIT
                        .ENDM

                        .MACRO  GETTIM  ARG             ;MACRO GET ELAPSED TIME
                                MOV     @#CLKCTR,ARG    ;STORE CLOCK COUNTER CONTENTS
                                CLR     @#CLKCSR        ;EVENT FINISHED, STOP CLOCK
                        .ENDM

                        .MACRO  STCLK                   ;MACRO START P-CLOCK
                                CLR     @#CLKCSB        ;CLEAR CLOCK COUNT SET BUFFER
                                CLR     @#CLKCTR        ;CLEAR CLOCK COUNTER
                                MOV     #23,@#
```

```
    52
    53                                  .NLIST  CND,MD,ME
    54
    55
    56  002000                                  POINTER BGNSW,BGNSFT,BGNDU
    57
    58  002000                          BGNMOD  MDHEDR
    63  002000                          HEADER  CZRLN,B,0,30000,0
   (4)  002000      103                         .ASCII /C/
   (4)  002001      132                         .ASCII /Z/
   (4)  002002      122                         .ASCII /R/
   (4)  002003      114                         .ASCII /L/
   (4)  002004      116                         .ASCII /N/
   (6)  002005      000                         .BYTE   0
   (6)  002006      000                         .BYTE   0
   (5)  002007      000                         .BYTE   0
   (4)  002010      102                         .ASCII  /B/
   (4)  002011      060                         .ASCII  /0/
   (4)  002012   000000                         .WORD   0
   (4)  002014   030000                         .WORD   30000
   (4)  002016   036352                         .WORD   L$HARD
   (4)  002020   036526                         .WORD   L$SOFT
   (4)  002022   013704                         .WORD   L$HW
   (4)  002024   013722                         .WORD   L$SW
   (4)  002026   037132                         .WORD   L$LAST
   (4)  002030   000000                         .WORD   0
   (4)  002032   000000                         .WORD   0
   (4)  002034   000000                         .WORD   0
   (4)  002036   000000                         .WORD   0
   (4)  002040   013740                         .WORD   L$DISPATCH
   (4)  002042   000000                         .WORD   0
   (4)  002044   000000                         .WORD   0
   (4)  002046   000000                         .WORD   0
   (4)  002050      003                         .BYTE   C$REVISION
   (3)  002051      003                         .BYTE   C$EDIT
   (4)  002052   000000                         .WORD   0
   (5)  002054   000000                         .WORD   0
   (4)  002056   000000                         .WORD   0
   (4)  002060   002216                         .WORD   L$DVTYP
   (4)  002062   000000                         .WORD   0
   (4)  002064   000000                         .WORD   0
   (4)  002066   000000                         .WORD   0
   (4)  002070   000000                         .WORD   0
   (4)  002072   015420                         .WORD   L$DU
   (4)  002074   000000                         .WORD   0
   (4)  002076   002122                         .WORD   L$DESC
   (4)  002100   104035                        EMT     E$LOAD
   (4)  002102   000000                         .WORD   0
   (4)  002104   013760                         .WORD   L$INIT
   (4)  002106   015272                         .WORD   L$CLEAN
   (4)  002110   014734                         .WORD   L$AUTO
   (4)  002112   013674                         .WORD   L$PROT
   (4)  002114   000000                         .WORD   0
   (4)  002116   000000                         .WORD   0
   (4)  002120   000000                         .WORD   0
   65   002122                         ENDMOD
```

```
 66  002122                                 DESCRIPT        <CZRLN TESTS SEEK & ROTATIONAL TIMING AND WRITE & READ DATA>
(3)  002122  055103  046122  020116         .ASCIZ  /CZRLN TESTS SEEK & ROTATIONAL TIMING AND WRITE & READ DATA/
(3)  002130  042524  052123  020123
(3)  002136  042523  045505  023040
(3)  002144  051040  052117  052101
(3)  002152  047511  040516  020114
(3)  002160  044524  044515  043516
(3)  002166  040440  042116  053440
(3)  002174  044522  042524  023040
(3)  002202  051040  040505  020104
(3)  002210  040504  040524     000
(2)          002216                         .EVEN
 67  002216                                 DEVTYP  <RL01,RL02>
(3)  002216  046122  030460  051054         .ASCIZ  /RL01,RL02/
(3)  002224  030114  000062
(2)                                         .EVEN
 68
 69                                         ;COPYRIGHT (C) 1979
 70                                         ;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
 71                                         ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
 72                                         ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
 73                                         ;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
 74                                         ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
 75                                         ;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
 76                                         ;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
 77                                         ;AT ALL TIMES REMAIN IN DEC.
 78                                         ;
 79                                         ;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
 80                                         ;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
 81                                         ;BY DIGITAL EQUIPMENT CORPORATION.
 82                                         ;
 83                                         ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
 84                                         ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
 85
 86
 87
 88                                     .SBTTL  GLOBAL DATA SECTION
 89
 90  002230                             BGNMOD  GLBEQAT
 91
 92  002230                             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)          ;
(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)          ;
(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
(1)          ;
(1)          ;OPERATOR FLAG BITS
(1)          ;
(1)  000004  EVL==      4
(1)  000010  LOT==     10
(1)  000020  ADR==     20
(1)  000040  IDU==     40
(1)  000100  ISR==    100
(1)  000200  WAM==    200
(1)  000400  BOE==    400
(1)  001000  PNT==   1000
(1)  002000  PRI==   2000
(1)  004000  IXE==   4000
(1)  010000  IBE==  10000
(1)  020000  IER==  20000
(1)  040000  LOE==  40000
(1)  100000  HOE== 100000
 93          ;     OFFSETS FOR HARDWARE P-TABLE
 94  000000  CSR   =0          ;BUS ADDRESS
 95  000002  VECT  =2          ;VECTOR ADDRESS
```

```
                                PRIOR   =4                      ;PRIORITY
                                TYPDR   =6                      ;DRIVE TYPE
                                DRSB    =10                     ;DRIVE SELECT BIT
                                CNT     =12                     ;CONTROLLER TYPE

                                ;       OFFSET FOR SOFTWARE P-TABLE
                                MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
                                LOLIM   =2                      ;CYLINDER LOWER LIMIT
                                HILIM   =4                      ;CYLINDER HIGH LIMIT
                                HEAD    -6                      ;SELECTED HEAD FOR RUNNING TESTS
                                ERLIM   =10                     ;ERROR LIMIT
                                DCLIM   =12                     ;DATA COMPARE ERROR LIMIT

                                ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
                                ALLCYL  =BIT00                  ;USE ALL CYLINDERS
                                ALLSEC  =BIT01                  ;USE ALL SECTORS
                                DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
                                HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
                                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 TOO LONG
```

```
                        NOIRPT  =BIT01                   ;NO INTERRUPT FROM OPERATION
                        CONHNG  =BIT02                   ;CONTROLLEP 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-EXISTENT MEMORY ERROR
                        DLTERR  =10000                   ;DATA LATE ERROR
                        HNFERR  =10000                   ;HEADER NOT FOUND ERROR
                        DCKERR  =4000                    ;DATA CHECK ERROR
                        HCRCERR =4000                    ;HEADER CHECK ERROR
                        OPIERR  =2000                    ;OPERATION INCOMPLETE ERROR
                        DSMSK   =1400                    ;DRIVE SELECT MASK
                        CRDYMSK =200                     ;CONTROLLER READY MASK
                        INTEBL  =100                     ;INTERRUPT ENABLE MASK
                        BAMSK   =60                      ;BUS ADDRESS UPPER MASK
                        DRDYMSK =1                       ;DRIVE READY MASK

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

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

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

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

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

                        ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
                        STAMSK  =7                       ;STATE MASK
                        BHSTAT  =10                      ;BRUSH HOME STATUS
                        HOSTAT  =20                      ;HEADS OUT STATUS
                        COSTAT  =40                      ;COVER OPEN STATUS
                        HSSTAT  =100                     ;HEAD SELECT STATUS
                        DSESTAT =400                     ;DRIVE SELECT ERROR STATUS
                        VCSTAT  =1000                    ;VOLUME CHECK STATUS
                        WGESTAT =2000                    ;WRITE GATE ERROR STATUS
                        SPDSTAT =4000                    ;SPIN ERROR STATUS
```

```
                        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

                                        ;       P-CLOCK REGISTERS
                        172540          CLKCSR  =172540                 ;CLOCK CONTROL AND STATUS REGISTER
                        172542          CLKCSB  =172542                 ;CLOCK COUNT SET BUFFER
                        172544          CLKCTR  =172544                 ;CLOCK COUNTER

002230                                  ENDMOD


                                        .SBTTL  GLOBAL DATA SECTION

002230                                  BGNMOD  GLBDAT

                                        ;       TABLE OF OPERATION MESSAGES

002230  000000                          OPMSGS: .WORD   0               ;FILLER
002232  005375                                  .WORD   MWRCHK          ;MESSAGE FOR WRITE CHECK
002234  005420                                  .WORD   MGTSTA          ;               GET STATUS
002236  005350                                  .WORD   MSEEK           ;               SEEK
002240  005365                                  .WORD   MREADH          ;               READ HEADER
002242  005406                                  .WORD   MWRITE          ;               WRITE DATA
002244  005354                                  .WORD   MREAD           ;               READ DATA
002246  005503                                  .WORD   MWRSET          ;               WITH RESET
002250  005432                                  .WORD   MDATCP          ;               WITH DATA COMPARE
002252  005451                                  .WORD   MHDRCP          ;               WITH HEADER COMPARE
002254  005550                                  .WORD   MCYLUP          ;               LOAD HEADS
002256  005537                                  .WORD   MULOAD          ;               UNLOAD HEADS
002260  005577                                  .WORD   MINOUT          ;                  IN-OUT SEQ
002262  005560                                  .WORD   MOUTIN          ;                  OUT-IN SEQ
002264  005620                                  .WORD   MFOLWRT         ;                  FOLLOWING WRITE
002266  005640                                  .WORD   MREVSK          ;                  REV SEEK
002270  005671                                  .WORD   MFWDSK          ;                  FWD SEEK
002272  005756                                  .WORD   MRESKO          ;                  REV SEEK
002274  005722                                  .WORD   MFWSKO          ;                  FWD SEEK
002276  006012                                  .WORD   MBADAD          ;                  BAD DISK ADD FOR WRITE
002300  005467                                  .WORD   M40HDR          ;               40 HEADER OPERATION
002302  000000                  T.DRIVE: .WORD  0
002304  000000                  JJJ:     .WORD  0
002306  000000                  HLMTW:   .WORD  0
002310  000000                  CLRBYT:  .WORD  0
002312  000000                  NXTHL:   .WORD  0
002314  000000                  GBND:    .WORD  0
002316  000000                  CAMSK:   .WORD  0
002320  000000                  DIRMSK:  .WORD  0
002322  000000                  HDCYL:   .WORD  0

                                        ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
002324  010135                  RESTBL: .WORD   MCERR            ;CONTROLLER ERROR
002326  010246                          .WORD   MDRERR          ;DRIVE ERROR
002330  010464                          .WORD   MNEERR          ;NON-EXISTANT MEMORY ERROR
002332  010436                          .WORD   MFLERR          ;HEADER NOT FOUND-DATA LATE
```

```
                                        .WORD   MHDERR          ;HEADER OR DATA ERROR
                                        .WORD   MOPERR          ;OPERATION INCOMPLETE
                                        .WORD   MNDRST          ;NO DRIVE STATUS AVAILABLE
                                        .WORD   0
                                        .WORD   MWDERR          ;WRITE DATA ERROR
                                        .WORD   MHCERR          ;HEAD CURRENT ERROR
                                        .WORD   0
                                        .WORD   MSTERR          ;SEEK TIMEOUT ERROR
                                        .WORD   MSPERR          ;SPINDLE ERROR
                                        .WORD   MWGERR          ;WRITE GATE ERROR
                                        .WORD   0
                                        .WORD   MDSERR          ;DRIVE SELECT ERROR

                                ;       PATTERN TABLE
                                PATTBL: .WORD   PAT1
                                        .WORD   PAT2
                                        .WORD   PAT3
                                        .WORD   PAT4
                                        .WORD   PAT5
                                        .WORD   PAT6
                                        .WORD   PAT7
                                        .WORD   PAT8
                                        .WORD   PAT9
                                        .WORD   PAT10


                                ;       SUBROUTINE CALLING STACK
                                SUBSTK: .WORD   0               ;STACK IS 12 WORDS LONG
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0


                                ;RL01 TABLE OF CYLINDERS
                                T25TBL: .WORD   2               ;TABLE OF DIFFERENCES
                                        .WORD   6
                                        .WORD   9.
                                        .WORD   12.
                                        .WORD   17.
                                        .WORD   22.
                                        .WORD   27.
                                        .WORD   34.
                                        .WORD   41.
                                        .WORD   128.
                                        .WORD   255.

                                ;RL02 TABLE OF CYLINDERS
                                T25T92: .WORD   4
                                        .WORD   12.
                                        .WORD   18.
                                        .WORD   24.
```

```
  320  002472  000042                    .WORD   34.
  321  002474  000054                    .WORD   44.
  322  002476  000066                    .WORD   54.
  323  002500  000104                    .WORD   68.
  324  002502  000122                    .WORD   82.
  325  002504  000400                    .WORD   256.
  326  002506  000777                    .WORD   511.
  327
  328                          ;       TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS
  329
  330  002510  000020          T33TBL: .BLKW   16.
  331  002550  000020          TBT:    .BLKW   16.
  332
  333
  334  002610      002          CYLTBL: .BYTE   2               ;TABLE OF DEFAULT CYLINDERS
  335  002611      007                  .BYTE   7.
  336  002612      016                  .BYTE   14.
  337  002613      024                  .BYTE   20.
  338  002614      033                  .BYTE   27.
  339  002615      041                  .BYTE   33.
  340  002616      046                  .BYTE   38.
  341  002617      055                  .BYTE   45.
  342  002620      064                  .BYTE   52.
  343  002621      072                  .BYTE   58.
  344  002622      101                  .BYTE   65
  345  002623      110                  .BYTE   72.
  346  002624      115                  .BYTE   77.
  347  002625      124                  .BYTE   84.
  348  002626      133                  .BYTE   91.
  349  002627      141                  .BYTE   97.
  350  002630      146                  .BYTE   102.
  351  002631      154                  .BYTE   108.
  352  002632      161                  .BYTE   113.
  353  002633      170                  .BYTE   120.
  354  002634      177                  .BYTE   127.
  355  002635      206                  .BYTE   134.
  356  002636      213                  .BYTE   139.
  357  002637      222                  .BYTE   146.
  358  002640      230                  .BYTE   152.
  359  002641      235                  .BYTE   157.
  360  002642      244                  .BYTE   164.
  361  002643      252                  .BYTE   170.
  362  002644      261                  .BYTE   177.
  363  002645      270                  .BYTE   184.
  364  002646      275                  .BYTE   189.
  365  002647      303                  .BYTE   195.
  366  002650      312                  .BYTE   202.
  367  002651      317                  .BYTE   207.
  368  002652      326                  .BYTE   214.
  369  002653      334                  .BYTE   220.
  370  002654      343                  .BYTE   227.
  371  002655      352                  .BYTE   234.
  372  002656      361                  .BYTE   241.
  373  002657      367                  .BYTE   247.
  374  002660      375                  .BYTE   253.
  375* 002661      000                  .BYTE   0
```

```
376  002662  000401                          .WORD   257.
377  002664  000406                          .WORD   262.
378  002666  000415                          .WORD   269.
379  002670  000423                          .WORD   275.
380  002672  000432                          .WORD   282.
381  002674  000445                          .WORD   293.
382  002676  000454                          .WORD   300.
383  002700  000463                          .WORD   307.
384  002702  000471                          .WORD   313.
385  002704  000500                          .WORD   320.
386  002706  000507                          .WORD   327.
387  002710  000514                          .WORD   332.
388  002712  000523                          .WORD   339.
389  002714  000532                          .WORD   346.
390  002716  000540                          .WORD   352.
391  002720  000545                          .WORD   357.
392  002722  000553                          .WORD   363.
393  002724  000560                          .WORD   368.
394  002726  000567                          .WORD   375.
395  002730  000576                          .WORD   382.
396  002732  000605                          .WORD   389.
397  002734  000612                          .WORD   394.
398  002736  000621                          .WORD   401.
399  002740  000627                          .WORD   407.
400  002742  000634                          .WORD   412.
401  002744  000643                          .WORD   419.
402  002746  000651                          .WORD   425.
403  002750  000660                          .WORD   432.
404  002752  000667                          .WORD   439.
405  002754  000674                          .WORD   444.
406  002756  000702                          .WORD   450.
407  002760  000711                          .WORD   457.
408  002762  000716                          .WORD   462.
409  002764  000725                          .WORD   469.
410  002766  000733                          .WORD   475.
411  002770  000742                          .WORD   482.
412  002772  000751                          .WORD   489.
413  002774  000760                          .WORD   496.
414  002776  000766                          .WORD   502.
415  003000  000774                          .WORD   508.
416  003002  000774                          .WORD   508.
417  003004  000000                          .WORD   0
418  003006  000000              SSINDX:     .WORD   0           ;SUBROUTINE STACK INDEX POINTER
419
420                              ;           OPERATIONAL FLAGS
421  003010  000000              OPFLAG:     .WORD   0           ;OPERATION FLAGS
422  003012  000000              DONE:       .WORD   0           ;OPERATION COMPLETE FLAG
423  003014  000000              HADONE:     .WORD   0           ;HEAD ALIGNMENT DONE FLAG
424  003016  000000              ERHEAD:     .WORD   0           ;ADDRESS OF ERROR HEADER
425  003020  000000              MORECE:     .WORD   0           ;MORE THAN 1 COMPARE ERROR
426  003022  000000              ERRSWI:     .WORD   0           ;ERROR RETURN SWITCH
427  003024  000000              BSFLAG:     .WORD   0           ;BAD SECTOR FLAGS
428  003026  000000              WRTSWI:     .WORD   0           ;WRITE SWITCH
429  003030  000000              TBLSTR:     .WORD   0           ;TABLE STORAGE
430
431  003032  000000              RLBAS:      .WORD   0           ;RL11 BASE ADDRESS
```

```
003034  000000          RLVEC:   .WORD   0          ;RL11 VECTOR ADDRESS
003036  000000          RLDRV:   .WORD   0          ;DRIVE NUMBER UNDER TEST

003040  000000          L.CS:    .WORD   0          ;CONTROLLER REGISTER STORAGE
003042  000000          L.BA:    .WORD   0          ;BEFORE OPERATION
003044  000000          L.DA:    .WORD   0
003046  000000          L.MP:    .WORD   0
003050  000000          T.CS:    .WORD   0          ;CONTROLLER REGISTER STORAGE
003052  000000          T.BA:    .WORD   0          ;   AFTER OPERATION
003054  000000          T.DA:    .WORD   0
003056                  T.MP:
003056  000000          HDWRD1:  .WORD   0          ;HEADER WORD STORAGE
003060  000000          HDWRD2:  .WORD   0
003062  000000          HDWRD3:  .WORD   0

003064  000000          T.STAT:  .WORD   0          ;DRIVE STATE STORAGE

003066  000000          RESPARM: .WORD   0          ;PARAM BLOCK FOR REASON REPORT
003070  000000                   .WORD   0
003072  000000                   .WORD   0
003074  000000                   .WORD   0
003076  000000                   .WORD   0

003100  000000          DRVCNT:  .WORD   0          ;DRIVE COUNT FOR DRIVES UNDER TEST
003102  000000          DIFAUG:  .WORD   0          ;DIFFERENCE AUGMENT FOR SEEK
003104  000000          OLDCYL:  .WORD   0          ;OLD CYLINDER
003106  000000          NEWCYL:  .WORD   0          ;NEW CYLINDER
003110  000000          CURCYL:  .WORD   0          ;CURRENT CYLINDER
003112  000000          DESDIF:  .WORD   0          ;DESIRED DIFFERENCE
003114  000000          DESSGN:  .WORD   0          ;DESIRED SIGN
003116  000000          DESHD:   .WORD   0          ;DESIRED HEAD
003120  000000          DESSEC:  .WORD   0          ;DESIRED SECTOR
003122  000000          TEMP0:   .WORD   0          ;TEMPORARY STORAGE
003124  000000          TEMP1:   .WORD   0          ;TEMPORARY STARAGE
003126  000000          TEMP2:   .WORD   0          ;TEMPORARY STORAGE
003130  000000          TEMP3:   .WORD   0          ;TEMPORARY STORAGE
003132  000000          TEMP4:   .WORD   0          ;TEMPORARY STORAGE
003134  000000          TEMP5:   .WORD   0          ;TEMPORARY STORAGE
003136  000000          TEMP6:   .WORD   0          ;TEMPORARY STORAGE
003140  000000          TEMP7:   .WORD   0          ;TEMPORARY STORAGE
003142  000000          TEMP8:   .WORD   0          ;TEMPORARY STORAGE
                        ;        TIMER STORAGE
003144  000000          OFIN:    .WORD   0          ;ONE CYLINDER FORWARD INNER
003146  000000          OFINU:   .WORD   0          ;         UPPER
003150  000000          OFMID:   .WORD   0          ;ONE CYLINDER FORWARD MIDDLE
003152  000000          OFMIDU:  .WORD   0          ;         UPPER
003154  000000          OFOUT:   .WORD   0          ;ONE CYLINDER FORWARD OUTER
003156  000000          OFOUTU:  .WORD   0          ;         UPPER
003160  000000          ORIN:    .WORD   0          ;ONE CYLINDER REVERSE INNER
003162  000000          ORINU:   .WORD   0          ;         UPPER
003164  000000          ORMID:   .WORD   0          ;ONE CYLINDER REVERSE MIDDLE
003166  000000          ORMIDU:  .WORD   0          ;         UPPER
003170  000000          OROUT:   .WORD   0          ;ONE CYLINDER REVERSE OUTER
003172  000000          OROUTU:  .WORD   0          ;         UPPER
003174  000000          HFIN:    .WORD   0          ;128 CYLINDER FORWARD INNER
003176  000000          HFINU:   .WORD   0          ;         UPPER
```

```
                                HFOUT:   .WORD   0              ;128 CYLINDER FORWARD OUTER
                                HFOUTU:  .WORD   0              ;       UPPER
                                HRIN:    .WORD   0              ;128 CYLINDER REVERSE INNER
                                HRINU:   .WORD   0              ;       UPPER
                                HROUT:   .WORD   0              ;128 CYLINDER REVERSE OUTER
                                HROUTU:  .WORD   0              ;       UPPER
                                AFMID:   .WORD   0              ;256 CYLINDER FORWARD
                                AFMIDU:  .WORD   0              ;       UPPER
                                ARMID:   .WORD   0              ;256 CYLINDER REVERSE
                                ARMIDU:  .WORD   0              ;       UPPER

                                EXOCYL:  .WORD   150.           ;EXPECTED TIME ONE CYLINDER
                                EXHCYL:  .WORD   550.           ;EXPECTED TIME 128 CYLINDER
                                EXACYL:  .WORD   1000.          ;EXPECTED TIME 256 CYLINDER
                                EXROT:   .WORD   250.           ;EXPECTED ROTATION TIME
                                ERRVEC:  .WORD   4              ;ERROR VECTOR

                                ;       MISCELLANEOUS COUNTERS
                                PASCNT:  .WORD   0              ;PASS COUNTER (LOCAL TO A TEST)
                                COUNT:   .WORD   0              ;A COUNTER (LOCAL TO A TEST)
                                ERRPOINT: .WORD  0              ;ERROR POINTER
                                ERRCNT:  .BLKW   64.            ;ERROR COUNTER FOR PROGRAM
                                PASNUM:  .WORD   0              ;PASS NUMBER FOR PROGRAM
                                PSETNM:  .WORD   0              ;COUNTER FOR PARAMETER SET NUMBER IN USE
                                LOCERR:  .BYTE   0              ;LOCAL ERROR COUNTER
                                NOERCT:  .BYTE   0              ;INHIBIT ERROR COUNTING FLAG
                                TRPFLG:  .WORD   0              ;HARDWARE TRAP OCCURANCE
                                PWRFLG:  .WORD   0              ;POWER FAILURE OCCURANCE
                                XDELAY:  .WORD   0
                                YDELAY:  .WORD   0
                                MININC:  .WORD   0
                                TEMP:    .WORD   0
                                TIM.US:  .WORD   0
                                TAG:     .WORD   0
                                MAJINC:  .WORD   0
                                CLKFLG:  .WORD   0              ;FLAG INDICATING PRESENCE OF A P-CLOCK
                                CLKADR:  .WORD   0              ;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE


                                ;       BAD SECTOR TABLES AND POINTERS
                                BSFVAL:  .WORD   0              ;BAD SECTORS FILES VALID FLAG

                                SBSFIL:  .BLKW   76             ;SOFTWARE BAD SECTOR FILE
                                FBSFIL:  .BLKW   76             ;FACTORY BAD SECTOR FILE

                                IBUFF:   .BLKW   200            ;INPUT BUFFER
                                OBUFF:   .BLKW   200            ;OUTPUT BUFFER

                                PAT1:    .WORD   0              ;PATTERN 1 (ALL ZEROS)
                                PAT2:    .WORD   177772
                                         .WORD   177777
                                         .WORD   177777
                                         .WORD   052525
                                         .WORD   052525
                                         .WORD   052525
                                         .WORD   177777
```

```
  546  005112  177777                        .WORD   177777
  547  005114  052525                        .WORD   052525
  548  005116  052525                        .WORD   052525
  549  005120  177777                        .WORD   177777
  550  005122  052525                        .WORD   052525
  551  005124  177252                        .WORD   177252
  552  005126  177252                        .WORD   177252
  553  005130  172765                        .WORD   172765
  554  005132  172765                        .WORD   172765
  555
  556  005134  000003              PAT3:     .WORD   000003
  557  005136  000000                        .WORD   000000
  558  005140  000000                        .WORD   000000
  559  005142  177777                        .WORD   177777
  560  005144  177777                        .WORD   177777
  561  005146  177777                        .WORD   177777
  562  005150  000000                        .WORD   000000
  563  005152  000000                        .WORD   000000
  564  005154  177777                        .WORD   177777
  565  005156  177777                        .WORD   177777
  566  005160  000000                        .WORD   000000
  567  005162  177777                        .WORD   177777
  568  005164  000000                        .WORD   000000
  569  005166  177777                        .WORD   177777
  570  005170  000000                        .WORD   000000
  571  005172  177777                        .WORD   177777
  572
  573  005174  025252              PAT4:     .WORD   025252
  574  005176  052525                        .WORD   052525
  575  005200  052525                        .WORD   052525
  576  005202  125252                        .WORD   125252
  577  005204  125252                        .WORD   125252
  578  005206  125252                        .WORD   125252
  579  005210  052525                        .WORD   052525
  580  005212  052525                        .WORD   052525
  581  005214  125252                        .WORD   125252
  582  005216  125252                        .WORD   125252
  583  005220  052525                        .WORD   052525
  584  005222  125252                        .WORD   125252
  585  005224  052525                        .WORD   052525
  586  005226  125252                        .WORD   125252
  587  005230  052525                        .WORD   052525
  588  005232  125252                        .WORD   125252
  589
  590  005234  155555              PAT5:     .WORD   155555
  591  005236  133333                        .WORD   133333
  592  005240  066666                        .WORD   066666
  593
  594  005242  121105              PAT6:     .WORD   121105
  595  005244  150442                        .WORD   150442
  596  005246  064221                        .WORD   064221
  597  005250  132110                        .WORD   132110
  598  005252  055044                        .WORD   055044
  599  005254  026442                        .WORD   026442
  600  005256  013211                        .WORD   013211
  601  005260  105504                        .WORD   105504
```

```
602  005262  042642                          .WORD   042642
603  005264  021321                          .WORD   021321
604  005266  110550                          .WORD   110550
605  005270  044264                          .WORD   044264
606  005272  022132                          .WORD   022132
607  005274  011055                          .WORD   011055
608  005276  104426                          .WORD   104426
609  005300  042213                          .WORD   042213
610
611  005302  177777              PAT7:       .WORD   177777
612
613  005304  045513              PAT8:       .WORD   045513
614  005306  122645                          .WORD   122645
615  005310  151322                          .WORD   151322
616  005312  064551                          .WORD   064551
617  005314  132264                          .WORD   132264
618  005316  055132                          .WORD   055132
619  005320  026455                          .WORD   026455
620  005322  113226                          .WORD   113226
621  005324  045513                          .WORD   045513
622  005326  122645                          .WORD   122645
623  005330  151322                          .WORD   151322
624  005332  064551                          .WORD   064551
625  005334  132264                          .WORD   132264
626  005336  055132                          .WORD   055132
627  005340  026455                          .WORD   026455
628  005342  113226                          .WORD   113226
629
630  005344  125252              PAT9:       .WORD   125252
631
632  005346  155555              PAT10:      .WORD   155555
633
634  005350                      ENDMOD
635
636
637
641
642
643                              .SBTTL  GLOBAL MESSAGES
644
645  005350                      BGNMOD  GLBTXT
646
647  005350  045523  000040      MSEEK:  .ASCIZ  /SK /
648  005354  042122  042040  052101  MREAD:  .ASCIZ  /RD DATA /
649  005365     122  020104  042110  MREADH: .ASCIZ  /RD HDR /
650  005375     127  052122  041440  MWRCHK: .ASCIZ  /WRT CHCK/
651  005406  051127  020124  040504  MWRITE: .ASCIZ  /WRT DATA /
652  005420  042507  020124  052123  MGTSTA: .ASCIZ  /GET STAT /
653  005432  044527  044124  042040  MDATCP: .ASCIZ  /WITH DATA CMP /
654  005451     127  052111  020110  MHDRCP: .ASCIZ  /WITH HDR CMP /
655  005467     106  051117  032040  M40HDR: .ASCIZ  /FOR 40 HDRS/
656  005503     127  052111  020110  MWRSET: .ASCIZ  /WITH RESET /
657  005517     117  042520  035122  MOPER:  .ASCIZ  /OPER: /
658  005526  042522  052523  052114  MRSLT:  .ASCIZ  /RESULT:       /
659  005537     125  046116  020104  MULOAD: .ASCIZ  /UNLD DRV/
660  005550  042114  042040  053122  MCYLUP: .ASCIZ  /LD DRV /
```

```
   661  005560  047506  020114  020060  MOUTIN: .ASCIZ  /FOL 0 TO CC SK/
   662  005577     106  046117  031040  MINOUT: .ASCIZ  /FOL 255 TO CC SK/
   663  005620  047506  020114  051127  MFOLWRT: .ASCIZ /FOL WRT (NO SK)/
   664  005640  042101  020112  054503  MREVSK: .ASCIZ  /ADJ CYL WRTTN AFT REV SK/
   665  005671     101  045104  041440  MFWDSK: .ASCIZ  /ADJ CYL WRTTN AFT FWD SK/
   666  005722  045523  043040  042127  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
   667  005756  045523  051040  053105  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
   668  006012  047117  041040  042101  MBADAD: .ASCIZ  /ON BAD SEC FILES/
   669  006033     103  047101  052047  MBADSF: .ASCIZ  /CAN'T GET BAD SEC FILES/
   670  006063     102  042101  051440  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
   671  006110  047524  046440  047101  MTMBS:  .ASCIZ  /TO MANY BAD SEC /
   672  006131     102  051525  040440  BASADD: .ASCIZ  /BUS ADD=/
   673  006142  051104  036526     000  DRVNAM: .ASCIZ  /DRV=/
   674  006147     116  020117  051104  DRVNAV: .ASCIZ  /NO DRV FOR TST/
   675  006166  051104  020126  044504  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
   676  006226  046122  051503     000  CSNAM:  .ASCIZ  /RLCS/
   677  006233     122  041114  000101  BANAM:  .ASCIZ  /RLBA/
   678  006240  046122  040504     000  DANAM:  .ASCIZ  /RLDA/
   679  006245     122  046514  000120  MPNAM:  .ASCIZ  /RLMP/
   680  006252  050117  044440  044516  LAB1:   .ASCIZ  /OP INIT = /
   681  006265     117  020120  047504  LAB2:   .ASCIZ  /OP DONE = /
   682  006300  047527  042122  000040  MWORD:  .ASCIZ  /WORD /
   683  006306  047111  051124  052120  MTOSLOW: .ASCIZ /INTRPT TOO LATE/
   684  006326  047516  042040  053122  MDRRES: .ASCIZ  /NO DRV RSPNSE/
   685  006344  047516  044440  052116  MNOINT: .ASCIZ  /NO INTRPT ON CMND COMPLETE/
   686  006377     103  052116  051114  MCONHNG: .ASCIZ /CNTLR HUNG /
   687  006413     105  051122  042040  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
   688  006433     126  046117  041440  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
   689  006454  047125  050130  052103  UNXERR: .ASCIZ  /UNXPCTED ERR/
   690  006471     040  042524  052123  TSTLAB: .ASCIZ  / TEST/
   708  006477     117  052125  043440  P2T03E: .ASCIZ  /OUT GRD BAND /
   709  006515     111  041516  051440  P2T04E: .ASCIZ  /INC SK FWD HD 0/
   710  006535     111  041516  051440  P2T05E: .ASCIZ  /INC SK REV HD 0/
   711  006555     111  041516  051440  P2T06E: .ASCIZ  /INC SK FWD HD 1/
   712  006575     111  047116  043440  P2T07E: .ASCIZ  /INN GRD BAND /
   713  006613     111  041516  051440  P2T08E: .ASCIZ  /INC SK REV HD 1/
   714  006633     123  000113          P2T09E: .ASCIZ  /SK/
   715  006636  053506  020104  051517  P2T10E: .ASCIZ  /FWD OSC SK/
   716  006651     122  053105  047440  P2T11E: .ASCIZ  /REV OSC SK/
   717  006664  045523  052040  046511  P2T12E: .ASCIZ  /SK TIMING/
   718  006676  051502  020103  042122  P2T13E: .ASCIZ  /BSC RD DATA/
   719  006712  051127  027524  042122  P2T14E: .ASCIZ  &WRT/RD DATA (P1)&
   720  006733     123  044520  042116  P2T15E: .ASCIZ  /SPINDLE ROT TIMING/
   721  006756  051127  027524  042122  P2T16E: .ASCIZ  &WRT/RD DATA (P2)&
   722  006777     127  052122  046040  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROT/
   723  007031     101  045104  041440  P2T18E: .ASCIZ  /ADJ CYL INTERFNCE/
   724  007053     117  042526  053522  P2T19E: .ASCIZ  /OVERWRT/
   725  007063     123  020113  044524  SKTMES: .ASCIZ  /SK TIMES /
   726  007075     123  044520  042116  SRTMES: .ASCIZ  /SPINDLE ROT TIME /
   727  007117     050  047111  030440  VALDES: .ASCIZ  /(IN 100'S OF U-SEC)/
   728  007143     101  050120  047522  MAPROX: .ASCIZ  /APPROX /
   729  007153     111  047116  051105  LABIN:  .ASCIZ  /INNER/
   730  007161     115  042111  046104  LABMID: .ASCIZ  /MIDDLE/
   731  007170  052517  042524  000122  LABOUT: .ASCIZ  /OUTER/
   732  007176  040515  020130  044524  LABEXP: .ASCIZ  /MAX TIME/
   733  007207     061  041440  046131  LABOCF: .ASCIZ  /1 CYL FWD/
```

```
   734  007221     061  041440  046131  LABOCR: .ASCIZ  /1 CYL REV/
   735  007233     115  042111  041440  LABHCF: .ASCIZ  /MID CYL FWD/
   736  007247     115  042111  041440  LABHCR: .ASCIZ  /MID CYL REV/
   737  007263     115  054101  041440  LABACF: .ASCIZ  /MAX CYL FWD/
   738  007277     115  054101  041440  LABACR: .ASCIZ  /MAX CYL REV/
   740  007313     110  051504  043040  HDMOVF: .ASCIZ  /HDS FAILED TO MV IN 10 TRYS/
   758  007347     122  051505  052105  OPR12:  .ASCIZ  /RESET WRT LCK /
   759  007366  047117  000040          OPR1A:  .ASCIZ  /ON /
   760  007372  047117  042040  053122  OPR1B:  .ASCIZ  /ON DRV /
   761  007402  047125  042504  020122  UNDTST: .ASCIZ  /UNDER TEST/
   762  007415     123  052105  053440  OPR004: .ASCIZ  /SET WRT LCK /
   763  007432  044504  043106  000040  DIFWD:  .ASCIZ  /DIFF /
   764  007440  043523  020116     000  SGNWD:  .ASCIZ  /SGN /
   765  007445     110  020104     000  HDWD:   .ASCIZ  /HD /
   766  007451     123  041505  000040  SECWD:  .ASCIZ  /SEC /
   767  007456  054503  020114     000  CYLWD:  .ASCIZ  /CYL /
   768  007463     106  047522  020115  FRMWD:  .ASCIZ  /FROM /
   769  007471     040  054502  040520  BYPSNM: .ASCIZ  / BYPASSED /
   770  007504  047522  052125  047111  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ:/
   771  007527     104  053122  051440  STAMES: .ASCIZ  /DRV STAT/
   772  007540  040502  020104  042523  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
   773  007614  047524  040524  020114  TCERR:  .ASCIZ  /TOTAL CMP ERRS: /
   774  007635     104  044522  042526  NOCTLR: .ASCIZ  /DRIVE DROPPED - NO CONTROLLER/
   775  007673     104  044522  042526  NOTRDY: .ASCIZ  /DRIVE DROPPED - DID NOT RESPOND WITH 'READY'/
   776  007750  042524  052123  030440  NOTST1: .ASCIZ  /TEST 1 CANNOT BE PERFORMED...P-CLOCK IS NOT AVAILABLE/
   777  010036  042524  052123  032040  NOTST4: .ASCIZ  /TEST 4 CANNOT BE PERFORMED...P-CLOCK IS NOT AVAILABLE/
   778
   779
   780                                  ;       RESULT NAMES
   781  010124  051104  020126  042122  MDRDY:  .ASCIZ  /DRV RDY /
   782  010135     103  047117  020124  MCERR:  .ASCIZ  /CONT ERR /
   783  010147     110  051104  041440  MHCRC:  .ASCIZ  /HDR CRC/
   784  010157     104  052101  020101  MDCRC:  .ASCIZ  /DATA CRC/
   785  010170  042110  020122  047516  MHNF:   .ASCIZ  /HDR NOT FND/
   786  010204  040504  040524  046040  MDLT:   .ASCIZ  /DATA LATE/
   787  010216  042110  020122  047516  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
   788  010246  051104  020126  051105  MDRERR: .ASCIZ  /DRV ERR /
   797  010257     104  053122  051440  MDSERR: .ASCIZ  /DRV SEL ERR /
   798  010274  051104  020126  052123  MDRVST: .ASCIZ  /DRV STATE /
   799  010307     123  044520  020116  MSPERR: .ASCIZ  /SPIN TIMEOUT /
   800  010325     127  052122  043440  MWGERR: .ASCIZ  /WRT GAT ERR /
   801  010342  045523  052040  046511  MSTERR: .ASCIZ  /SK TIMEOUT /
   802  010356  042510  042101  041440  MHCERR: .ASCIZ  /HEAD CUR ERR /
   803  010374  051127  020124  040504  MWDERR: .ASCIZ  /WRT DAT ERR /
   804  010411     117  051120  044455  MOPERR: .ASCIZ  /OPR-INC/
   805  010421     110  051104  042057  MHDERR: .ASCIZ  &HDR/DAT ERR &
   806  010436  042110  020122  047516  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
   807  010464  047516  026516  054105  MNEERR: .ASCIZ  /NON-EXISTENT MEMORY /
   808  010511     103  046131  000040  MCYLOC: .ASCIZ  /CYL /
   809  010516  040503  023516  020124  MNDRST: .ASCIZ  /CAN'T GET DRV STAT/
   810  010541     125  045516  020116  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
   811  010606  040506  046111  052040  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLR/
   812  010645     127  052122  040440  MWRTAB: .ASCIZ  /WRT ABRTD/
   813  010657     040  053117  020122  MEXERS: .ASCIZ  / OVR ERR LIMIT - UNIT DRPPD /
   814  010714  042440  051122     000  MERRS:  .ASCIZ  / ERR/
   815  010721     207  177777     000  BELL:   .ASCIZ  <207><377><377>
```

```
816
817                                     ;
818  010725     111  020123     000     ;       RESULT SETTINGS
819  010731     040  041123  000040     RESE3:  .ASCIZ  /IS /
820                                     RESE4:  .ASCIZ  / SB /
821                                     ;       RESULT CONDITIONS
822  010736  044440  020116     000     RESE5:  .ASCIZ  / IN /
823  010743     040  043117  000040     RESE6:  .ASCIZ  / OF /
824  010750  052123  052101  020105     STATE2: .ASCIZ  /STATE 2/
825  010760  052123  052101  020105     STATE3: .ASCIZ  /STATE 3/
826  010770  052123  052101  020105     STATE5: .ASCIZ  /STATE 5/
830  011000  051461  020124  020063     C10MS:  .ASCIZ  /1ST 3 MS/
831  011011     065  030060  051515     C500MS: .ASCIZ  /500MS/
832  011017     103  041531  052440     CCYLUP: .ASCIZ  /CYC UP/
833  011026  040504  040524  054040     CAFDT:  .ASCIZ  /DATA XFR/
834  011037     065  051440  041505     C5SEC:  .ASCIZ  /5 SEC/
835
836  011045     045  022516  022524     FMTOP1: .ASCIZ  /%N%T%N%T%T%06%S%T%01%N/
837  011074  047045  052045  047445     FMTOP2: .ASCIZ  /%N%T%01%S1%T%01%N/
838  011116  047045  052045  047445     FMTOP3: .ASCIZ  /%N%T%01%S1%T%T%N/
839  011137     045  022524  000124     FMT1:   .ASCIZ  /%T%T/
840  011144  047045  052045  052045     FMT1.1: .ASCIZ  /%N%T%T/
841  011153     045  000124             FMT2:   .ASCIZ  /%T/
842  011156  047045     000             FMT3:   .ASCIZ  /%N/
843  011161     045  022516  022524     FMT4:   .ASCIZ  /%N%T%T%N/
844  011172  047045  052045  047445     FMT5:   .ASCIZ  /%N%T%06%S1%T%01/
845  011212  047045  051445  030461     FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
846  011254  047045  052045  047445     FMT7:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
847  011324  047045  052045  047445     FMT8:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
848  011356  047045  052045     000     FMT9:   .ASCIZ  /%N%T/
849  011363     045  022524  030517     FMT11:  .ASCIZ  /%T%01/
850  011371     045  022524  031517     FMT12:  .ASCIZ  /%T%03/
851  011377     045  022516  030523     FMT13:  .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
852  011443     045  022516  022524     FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
853  011475     045  022516  030523     FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
854  011531     045  022516  032523     FMT16:  .ASCIZ  /%N%S5%06/
855  011542  051445  030061  052045     FMT17:  .ASCIZ  /%S10%T%N%S11%06%N/
856  011564  047045  051445  032461     FMT18:  .ASCIZ  /%N%S15%T%S5%T%S4%T%S5%T%N/
857  011616  052045  051445  022464     FMT19:  .ASCIZ  /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
858  011653     045  022524  031123     FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
859  011703     045  022524  030523     FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
860  011726  047045  051445  030461     FMT22:  .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
861  011762  052045  052045  052045     FMT23:  .ASCIZ  /%T%T%T%01%N/
862  011776  047045  052045     000     FMT24:  .ASCIZ  /%N%T/
863  012003     045  022516  031104     FMT25:  .ASCIZ  /%N%D2%T/
864  012013     045  022516  030523     FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
865  012037     045  022516  022524     FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
866  012056  047045  052045  052045     FMT28:  .ASCIZ  /%N%T%T%T/
867
868  012067                             ENDMOD
869
874
```

```
                                        .SBTTL  ERROR MESSAGES
012070                                  BGNMOD  GLBERR
                                        ;       ERR1    R3 POINTS TO RESULT MESSAGE
                                        ;               RESULT: (R3)
                                        ;
                                        ;       ERR2    R3 POINTS TO RESULT NAME
                                        ;               RESULT: (R3) IS 1 SB 0
                                        ;
                                        ;       ERR3    R3 POINTS TO RESULT NAME
                                        ;               RESULT: (R3) IS 0 SB 1
                                        ;
                                        ;       ERR4    R3 POINTS TO RESULT NAME
                                        ;               R4 POINTS TO RESULT CONDITIONS
                                        ;               RESULT: (R3) IS 1 SB 0 (R4)
                                        ;
                                        ;       ERR5    R3 POINTS TO RESULT NAME
                                        ;               R4 POINTS TO RESULT CONDITIONS
                                        ;               RESULT: (R3) IS 0 SB 1 (R4)
                                        ;
                                        ;       ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
                                        ;               REPORTS ALL
                                        ;               RESULT: 'ERROR'' IS 1 SB 0
                                        ;
                                        ;       ERR7    DRIVE STATE ERROR REPORT
                                        ;               R3 CONTAINS EXPECTED STATE
                                        ;               T.STAT CONTAINS BAD STATE
                                        ;               RESULT: DRIVE STATE IS (T.STAT) SB (R3)
                                        ;
                                        ;       ERR8    HEAD POSITIONING ERROR REPORT
                                        ;               NEWCYL CONTAINS EXPECTED CYLINDER
                                        ;               HDWRD1 CONTAINS BAD CYLINDER
                                        ;               RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
                                        ;
                                        ;       ERR9    UTILITY RESULT REPORT
                                        ;               R3 POINTS TO RESU T 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)
012070                                  BGNMSG  ERR1
012070  105737  003451                         TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
012074  001002                                BNE     1$              ;YES - SKIP
012076  005277  171140                         INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
012102  010146                         1$:     MOV     R1,-(SP)        ;STORE R1
012104  004737  024662                         JSR     PC,RPTOP        ;REPORT OPERATION
012110  012721  000001                         MOV     #1,(R1)+        ;SET PARAM NUMBER
012114  010321                                MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
012116  004737  025450                         JSR     PC,RPTRES       ;REPORT RESULTS
012122  004737  025656                         JSR     PC,RPTREM       ;REPORT REMAINDER
```

```
                                                        MOV     (SP)+,R1        ;RESTORE R1
                                                        JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                                ENDMSG
                                                L10000:
                                                        TRAP    C$MSG

                                                BGNMSG  ERR2
                                                        INC     @ERRPOINT       ;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     #1,(R1)+        ;SET IS VALUE
                                                        CLR     (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
                                                L10001:
                                                        TRAP    C$MSG

                                                BGNMSG  ERR3
                                                        INC     @ERRPOINT       ;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
                                                        CLR     (R1)+           ;SET IS VALUE
                                                        MOV     #1,(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
                                                L10002:
                                                        TRAP    C$MSG

                                                BGNMSG  ERR4
                                                        INC     @ERRPOINT       ;BUMP ERROR COUNT
                                                        MOV     R1,-(SP)        ;STORE R1
                                                        JSR     PC,RPTOP        ;REPORT OPERATION
                                                        MOV     #4,(R1)+        ;SET PARAM NUMBER
                                                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                                        MOV     #1,(R1)+        ;SET IS VALUE
                                                        CLR     (R1)+           ;SET SB VALUE
                                                        MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
                                                        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
                                                L10003:
                                                        TRAP    C$MSG

                                                BGNMSG  ERR5
```

```
  980  012322  005277  170714                        INC     @ERRPOINT       ;BUMP ERROR COUNT
  981  012326  010146                                MOV     R1,-(SP)        ;STORE R1
  982  012330  004737  024662                        JSR     PC,RPTOP        ;REPORT OPERATION
  983  012334  012721  000004                        MOV     #4,(R1)+        ;SET PARAM NUMBER
  984  012340  010321                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
  985  012342  005021                                CLR     (R1)+           ;SET IS VALUE
  986  012344  012721  000001                        MOV     #1,(R1)+        ;SET SB VALUE
  987  012350  010411                                MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
  988  012352  004737  025450                        JSR     PC,RPTRES       ;REPORT RESULTS
  989  012356  004737  025656                        JSR     PC,RPTREM       ;REPORT REMAINDER
  990  012362  012601                                MOV     (SP)+,R1        ;RESTORE R1
  991  012364  004737  016032                        JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
  992  012370                              ENDMSG
  (3)  012370                              L10004:
  (3)  012370  104423                                TRAP    C$MSG
  993
  994  012372                              BGNMSG  ERR6
  995  012372  105737  003451                        TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
  996  012376  001002                                BNE     17$             ;YES - SKIP
  997  012400  005277  170636                        INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
  998  012404  010146                              17$:MOV     R1,-(SP)        ;STORE R1
  999  012406  010346                                MOV     R3,-(SP)        ;STORE R3
 1000  012410  010446                                MOV     R4,-(SP)        ;STORE R4
 1001  012412  010546                                MOV     R5,-(SP)        ;STORE R5
 1002  012414  004737  024662                        JSR     PC,RPTOP        ;REPORT OPERATION
 1003  012420  012721  000003                        MOV     #3,(R1)+        ;SET PARAM NUMBER
 1004  012424  012761  000001  000002                MOV     #1,2(R1)        ;INSERT IS VALUE
 1005  012432  005037  003130                        CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
 1006  012436  013703  003050                        MOV     T.CS,R3         ;GET T.CS
 1007  012442  042703  177761                        BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
 1008  012446  022703  000004                        CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
 1009  012452  001434                                BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
 1010  012454  012762  000003  000004                MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
 1011  012462  012703  000004                        MOV     #4,R3
 1012  012466  053703  003036                        BIS     RLDRV,R3
 1013  012472  010362  000000                        MOV     R3,RLCS(R2)
 1014  012476                                WAITUS  #10.                    ;WAIT FOR CONTROLLER READY
 1015  012510  032762  000200  000000                BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
 1016  012516  001003                                BNE     10$             ;YES - SKIP
 1017  012520  012703  001000                  9$:   MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
 1018  012524  000413                                BR      2$              ;IN MESSAGE WORD AND SKIP
 1019  012526  016203  000006                 10$:   MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
 1020  012532  010337  003130                        MOV     R3,TEMP3
 1021  012536  113703  003131                        MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
 1022  012542  000402                                BR      13$
 1023  012544  113703  003057                  1$:   MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
 1024  012550  042703  177442                 13$:   BIC     #177442,R3      ;CLEAR UNUSED BITS
 1025  012554  013704  003050                  2$:   MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
 1026  012560  042704  001777                        BIC     #1777,R4        ;CLEAR UNUSED BITS
 1027  012564  050403                                BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
 1028  012566  032703  002000                        BIT     #OPIERR,R3      ;TEST IF OPI SET
 1029  012572  001442                                BEQ     115$            ;NO - SKIP
 1030  012574  032703  010000                        BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
 1031  012600  001026                                BNE     107$            ;YES - SKIP
 1032  012602  032703  004000                        BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
 1033  012606  001020                                BNE     105$            ;YES - SKIP
```

```
 1034  012610  012704  010411                 MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
 1035  012614                          100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
  (10) 012614  012746  010714                 MOV     #MERRS,-(SP)
  (9)  012620  010446                         MOV     R4,-(SP)
  (8)  012622  012746  005526                 MOV     #MRSLT,-(SP)
  (7)  012626  012746  012056                 MOV     #FMT28,-(SP)
  (6)  012632  012746  000004                 MOV     #4,-(SP)
  (3)  012636  010600                         MOV     SP,R0
  (4)  012640  104414                         TRAP    C$PNTB
  (4)  012642  062706  000012                 ADD     #12,SP
 1036  012646  000430                         BR      120$            ;SKIP
 1037  012650  012704  010147          105$:   MOV     #MHCRC,R4       ;HDR CRC MESSAGE
 1038  012654  000757                         BR      100$
 1039  012656  032703  004000          107$:   BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
 1040  012662  001003                         BNE     109$            ;YES - SKIP
 1041  012664  012704  010170                 MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
 1042  012670  000751                         BR      100$
 1043  012672  012704  010216          109$:   MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
 1044  012676  000746                         BR      100$            ;SKIP
 1045  012700  032703  004000          115$:   BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
 1046  012704  001403                         BEQ     118$            ;NO - SKIP
 1047  012706  012704  010157                 MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
 1048  012712  000740                         BR      100$            ;SKIP
 1049  012714  032703  010000          118$:   BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
 1050  012720  001403                         BEQ     120$            ;NO - SKIP
 1051  012722  012704  010204                 MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
 1052  012726  000732                         BR      100$            ;SKIP
 1053  012730  012705  100000          120$:   MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
 1054  012734  005004                         CLR     R4              ;CLEAR R4 FOR TABLE COUNT
 1055  012736  030503                  3$:     BIT     R5,R3           ;TEST IF BIT IS SET
 1056  012740  001005                         BNE     6$              ;YES - SKIP TO REPORT
 1057  012742  005724                  4$:     TST     (R4)+           ;ELSE BUMP TABLE POINTER
 1058  012744  000241                         CLC                     ;CLEAR CARRY
 1059  012746  006005                         ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
 1060  012750  001372                         BNE     3$              ;LOOP IF NOT 0
 1061  012752  000405                         BR      7$              ;ELSE REPORT REMAINDER
 1062  012754  016411  002324          6$:     MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
 1063  012760  004737  025450                 JSR     PC,RPTRES       ;REPORT RESULTS
 1064  012764  000766                         BR      4$              ;GET NEXT BIT
 1065  012766  004737  025656          7$:     JSR     PC,RPTREM       ;REPORT REMAINDER
 1066  012772  005737  003130                 TST     TEMP3           ;TEST IF ANY NEW STATUS
 1067  012776  001414                         BEQ     15$             ;NO - SKIP
 1068  013000                                 PRINTB  #FMT17,#STAMES,TEMP3
  (9)  013000  013746  003130                 MOV     TEMP3,-(SP)
  (8)  013004  012746  007527                 MOV     #STAMES,-(SP)
  (7)  013010  012746  011542                 MOV     #FMT17,-(SP)
  (6)  013014  012746  000003                 MOV     #3,-(SP)
  (3)  013020  010600                         MOV     SP,R0
  (4)  013022  104414                         TRAP    C$PNTB
  (4)  013024  062706  000010                 ADD     #10,SP
 1069  013030  032737  004000  003050  15$:   BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
 1070  013036  001453                         BEQ     25$             ;NO - SKIP
 1071  013040  032737  002000  003050         BIT     #OPIERR,T.CS    ;TEST IF OPI SET
 1072  013046  001047                         BNE     25$             ;YES - SKIP
 1073  013050  005037  003020                 CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
 1074  013054  012701  000200                 MOV     #128.,R1        ;SET COMPARE LENGTH
```

```
 1075  013060  012703  000001                MOV     #1,R3           ;SET WORD COUNT
 1076  013064  012705  004472                MOV     #OBUFF,R5       ;SET GOOD WC   POINTER
 1077  013070  012704  004072                MOV     #IBUFF,R4       ;SET TEST WORD  OINTER
 1078  013074  021514              18$:      CMP     (R5),(R4)       ;CHECK WORD
 1079  013076  001427                        BEQ     19$             ;GOOD - SKIP
 1080  013100  023727  003020  000012        CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
 1081  013106  003021                        BGT     20$             ;YES - SKIP
 1082  013110                                PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
  (13) 013110  011546                        MOV     (R5),-(SP)
  (12) 013112  012746  010731                MOV     #RESE4,-(SP)
  (11) 013116  011446                        MOV     (R4),-(SP)
  (10) 013120  012746  010725                MOV     #RESE3,-(SP)
  (9)  013124  010346                        MOV     R3,-(SP)
  (8)  013126  012746  006300                MOV     #MWORD,-(SP)
  (7)  013132  012746  011475                MOV     #FMT15,-(SP)
  (6)  013136  012746  000007                MOV     #7,-(SP)
  (3)  013142  010600                        MOV     SP,R0
  (4)  013144  104414                        TRAP    C$PNTB
  (4)  013146  062706  000020                ADD     #20,SP
 1083  013152  005237  003020        20$:    INC     MORECE          ;BUMP ERROR COUNTER
 1084  013156  022524              19$:      CMP     (R5)+,(R4)+     ;BUMP POINTERS
 1085  013160  005203                        INC     R3              ;BUMP COUNTER
 1086  013162  005301                        DEC     R1              ;DEC LENGTH COUNT
 1087  013164  001343                        BNE     18$             ;LOOP IF NOT DONE
 1088  013166  005737  003020        25$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
 1089  013172  001421                        BEQ     27$             ;NO - SKIP
 1090  013174  012701  000200                MOV     #128.,R1        ;SET COMPARE LENGTH
 1091  013200                                PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
  (11) 013200  010146                        MOV     R1,-(SP)
  (10) 013202  012746  010743                MOV     #RESE6,-(SP)
  (9)  013206  013746  003020                MOV     MORECE,-(SP)
  (8)  013212  012746  007614                MOV     #TCERR,-(SP)
  (7)  013216  012746  012037                MOV     #FMT27,-(SP)
  (6)  013222  012746  000005                MOV     #5,-(SP)
  (3)  013226  010600                        MOV     SP,R0
  (4)  013230  104414                        TRAP    C$PNTB
  (4)  013232  062706  000014                ADD     #14,SP
 1092  013236  012605              27$:      MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
 1093  013240  012604                        MOV     (SP)+,R4
 1094  013242  012603                        MOV     (SP)+,R3
 1095  013244  012601                        MOV     (SP)+,R1
 1096  013246  004737  016032                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1097  013252                        ENDMSG
  (3)  013252                        L10005:
  (3)  013252  104423                        TRAP    C$MSG
 1098
 1099  013254                        BGNMSG  ERR7
 1100  013254  005277  167762                INC     @ERRPOINT       ;BUMP ERROR COUNT
 1101  013260  010146                        MOV     R1,-(SP)        ;STORE R1
 1102  013262  004737  024662                JSR     PC,RPTOP        ;REPORT OPERATION
 1103  013266  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
 1104  013272  012721  010274                MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
 1105  013276  013721  003064                MOV     T.STAT,(R1)+    ;INSERT IS VALUE
 1106  013302  010311                        MOV     R3,(R1) ;INSERT SB VALUE
 1107  013304  004737  025450                JSR     PC,RPTRES       ;REPORT RESULTS
 1108  013310  004737  025656                JSR     PC,RPTREM       ;REPORT REMAINDER
```

```
1109  013314  012601                          MOV     (SP)+,R1        ;RESTORE R1
1110  013316  004737  016032                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
1111  013322                          ENDMSG
 (3)  013322                          L10006:
 (3)  013322  104423                          TRAP    C$MSG
1112
1113  013324                          BGNMSG  ERR8
1114  013324  005277  167712                  INC     @ERRPOINT       ;BUMP ERROR COUNT
1115  013330  010146                          MOV     R1,-(SP)        ;STORE R1
1116  013332  010346                          MOV     R3,-(SP)        ;STORE R3
1117  013334  004737  024662                  JSR     PC,RPTOP        ;REPORT OPERATION
1118  013340  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
1119  013344  012721  010511                  MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
1120  013350  013711  003056                  MOV     HDWRD1,(R1)     ;GET HEADER WORD
1121  013354  012703  000007                  MOV     #7,R3           ;SET SHIFT COUNT
1122  013360  000241                  3$:     CLC
1123  013362  006011                          ROR     (R1)            ;ALIGN CHAR FOR PRINTING
1124  013364  005303                          DEC     R3              ;  AS IS VALUE
1125  013366  001374                          BNE     3$
1126  013370  005721                          TST     (R1)+           ;BUMP PARAM POINTER
1127  013372  013711  003106                  MOV     NEWCYL,(R1)     ;INSERT SB VALUE
1128  013376  004737  025450                  JSR     PC,RPTRES       ;REPORT RESULTS
1129  013402  004737  025656                  JSR     PC,RPTREM       ;REPORT REMAINDER
1130  013406  012603                          MOV     (SP)+,R3        ;RESTORE R3
1131  013410  012601                          MOV     (SP)+,R1        ;RESTORE R1
1132  013412  004737  016032                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
1133  013416                          ENDMSG
 (3)  013416                          L10007:
 (3)  013416  104423                          TRAP    C$MSG
1134
1135  013420                          BGNMSG  ERR9
1136  013420  005277  167616                  INC     @ERRPOINT       ;BUMP ERROR COUNT
1137  013424  010146                          MOV     R1,-(SP)        ;STORE R1
1138  013426  004737  024662                  JSR     PC,RPTOP        ;REPORT OPERATION
1139  013432  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
1140  013436  010321                          MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
1141  013440  010421                          MOV     R4,(R1)+        ;SET IS VALUE
1142  013442  010521                          MOV     R5,(R1)+        ;SET SB VALUE
1143  013444  004737  025450                  JSR     PC,RPTRES       ;REPORT RESULTS
1144  013450  004737  025656                  JSR     PC,RPTREM       ;REPORT REMAINDER
1145  013454  012601                          MOV     (SP)+,R1        ;RESTORE R1
1146  013456  004737  016032                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
1147  013462                          ENDMSG
 (3)  013462                          L10010:
 (3)  013462  104423                          TRAP    C$MSG
1148  013464                          BGNMSG  ERR10
1149  013464  010146                          MOV     R1,-(SP)        ;STORE R1
1150  013466  005737  003020                  TST     MORECE          ;TEST IF 2ND BAD LINE
1151  013472  001051                          BNE     3$              ;YES - SKIP
1152  013474  005277  167542                  INC     @ERRPOINT       ;BUMP ERROR COUNT
1153  013500  004737  024662                  JSR     PC,RPTOP        ;REPORT OPERATION
1154  013504                                  PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
 (11) 013504  005046                          CLR     -(SP)
 (11) 013506  153716  003037                  BISB    RLDRV+1,(SP)
 (10) 013512  012746  006142                  MOV     #DRVNAM,-(SP)
 (9)  013516  013746  003032                  MOV     RLBAS,-(SP)
```

```
   (8)  013522  012746  006131                MOV     #BASADD,-(SP)
   (7)  013526  012746  011172                MOV     #FMT5,-(SP)
   (6)  013532  012746  000005                MOV     #5,-(SP)
   (3)  013536  010600                        MOV     SP,R0
   (4)  013540  104414                        TRAP    C$PNTB
   (4)  013542  062706  000014                ADD     #14,SP
  1155  013546                                PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
  (14)  013546  011546                        MOV     (R5),-(SP)
  (13)  013550  012746  010731                MOV     #RESE4,-(SP)
  (12)  013554  011446                        MOV     (R4),-(SP)
  (11)  013556  012746  010725                MOV     #RESE3,-(SP)
  (10)  013562  010346                        MOV     R3,-(SP)
   (9)  013564  012746  006300                MOV     #MWORD,-(SP)
   (8)  013570  012746  005526                MOV     #MRSLT,-(SP)
   (7)  013574  012746  011443                MOV     #FMT14,-(SP)
   (6)  013600  012746  000010                MOV     #10,-(SP)
   (3)  013604  010600                        MOV     SP,R0
   (4)  013606  104414                        TRAP    C$PNTB
   (4)  013610  062706  000022                ADD     #22,SP
  1156  013614  000421                        BR      4$
  1157  013616                        3$:     PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
  (13)  013616  011546                        MOV     (R5),-(SP)
  (12)  013620  012746  010731                MOV     #RESE4,-(SP)
  (11)  013624  011446                        MOV     (R4),-(SP)
  (10)  013626  012746  010725                MOV     #RESE3,-(SP)
   (9)  013632  010346                        MOV     R3,-(SP)
   (8)  013634  012746  006300                MOV     #MWORD,-(SP)
   (7)  013640  012746  011475                MOV     #FMT15,-(SP)
   (6)  013644  012746  000007                MOV     #7,-(SP)
   (3)  013650  010600                        MOV     SP,R0
   (4)  013652  104414                        TRAP    C$PNTB
   (4)  013654  062706  000020                ADD     #20,SP
  1158  013660  005237  003020        4$:     INC     MORECE          ;INC COMPARE ERROR COUNT
  1159  013664  012601                        MOV     (SP)+,R1        ;RESTORE R1
  1160  013666  004737  016032                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
  1161  013672                        ENDMSG
   (3)  013672                        L10011:
   (3)  013672  104423                        TRAP    C$MSG
  1162  013674                        ENDMOD
  1163
  1164                                ;LOAD PROTECTION TABLE
  1165  013674                        BGNPROT
  1166  013674  000000                        .WORD    0              ;OFFSET OF CSR IN P-TABLE
  1167  013676  177777                        .WORD   -1              ;NOT A MASS-BUS DRIVE
  1168  013700  000010                        .WORD   10              ;OFFSET OF DRIVE IN P-TABLE
  1169  013702                        ENDPROT
  1170
  1171                                        .EVEN
  1172
  1173  013702                        BGNMOD  HPTCODE
  1174  013702                        BGNHW
   (3)  013702  000006                        .WORD   L10013-L$HW/2
  1175  013704  174400                        .WORD   174400          ;CSR BASE ADDRESS DEFAULT
  1176  013706  000160                        .WORD   160             ;VECTOR DEFAULT
  1177  013710  000240                        .WORD   240             ;PRIORITY DEFAULT
  1178  013712  000001                        .WORD   1               ;TYPE OF DRIVE
```

```
1179  013714  000000                    .WORD   0               ;DRIVE NUMBER DEFAULT
1180  013716  000001                    .WORD   1               ;RL11 CONTROLLER
1181  013720                    ENDHW
 (3)  013720                    L10013:
1182  013720                    ENDMOD
1183
1184  013720                    BGNMOD  SPTCODE
1185  013720                    BGNSW
 (3)  013720  000006                    .WORD   L10014-L$SW/2
1186  013722  000000            MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
1187                                                            ;BIT 1 = USE ALL SECTORS
1188                                                            ;BIT 2 = EXECUTE DRIVE SELECT TEST
1189                                                            ;BIT 3 = EXECUTE HEAD ALIGNMENT
1190                                                            ;BIT 12 = HEAD SELECT SUPPLIED FLAG
1191                                                            ;BIT 13 = HILIMIT SPECIFIED FLAG
1192                                                            ;BIT 14 = LO LIMIT SPECIFIED FLAG
1193                                                            ;BIT 15 = DO MANUAL INTERVENTION
1194  013724  000000            LOLIMW: .WORD   0
1195  013726  000377            HILIMW: .WORD   255.
1196  013730  000000            HEADW:  .WORD   0
1197  013732  000024            ERLIMW: .WORD   20.             ;ERROR LIMIT
1198  013734  000012            DCLIMW: .WORD   10.             ;COMPARE ERROR LIMIT
1199  013736                    ENDSW
 (3)  013736                    L10014:
1200  013736                    ENDMOD
1201
1202  013736                    BGNMOD  DSPCODE
1207  013736                    DISPATCH        8
 (4)  013736  000010                    .WORD   8
 (6)  013740  026142                    .WORD   T1
 (6)  013742  030054                    .WORD   T2
 (6)  013744  030572                    .WORD   T3
 (6)  013746  031006                    .WORD   T4
 (6)  013750  031614                    .WORD   T5
 (6)  013752  032724                    .WORD   T6
 (6)  013754  033742                    .WORD   T7
 (6)  013756  035156                    .WORD   T8
1209  013760                    ENDMOD
1210
1211
```

```
                                        .SBTTL  INITIALIZATION SECTION

                                        BGNMOD  INITCODE
013760                                  BGNINIT

                                        ;CHECK FOR PRESENCE OF A P-CLOCK
013760  005037  003474                          CLR     CLKFLG          ;CLEAR CLOCK FLAG
013764                                          CLOCK   P,CLKADR        ;P-CLOCK?
013764  012700  000120                          MOV     #'P,R0
013770  104462                                  TRAP    C$CLCK
013772  010037  003476                          MOV     R0,CLKADR
013776                                          BNCOMPLETE      1$      ;BRANCH IF NO P-CLOCK
013776  103002                                  BCC     1$
014000  005237  003474                          INC     CLKFLG          ;INDICATE PRESENCE OF A P-CLOCK
014004                                  1$:     SETPRI  #340            ;SET PRIORITY TO 7 TO INHIBIT ALL INTERRUPTS
014004  012700  000340                          MOV     #340,R0
014010  104441                                  TRAP    C$SPRI
014012                                          BRESET                  ;FOR LSI-11 CPU'S
014012  104433                                  TRAP    C$RESET
014014                                          MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
014014  104450                                  TRAP    C$MANI
014016                                          BCOMPLETE       2$      ;YES - SKIP
014016  103403                                  BCS     2$
014020  042737  100014  013722                  BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
                                                                        ; INTERVENTION FLAGS
014026  005037  003006                  2$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
014032                                          READEF  #EF.PWR         ;POWER FAILURE
014032  012700  000034                          MOV     #EF.PWR,R0
014036  104447                                  TRAP    C$REFG
014040                                          BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
014040  103005                                  BCC     4$
014042  013737  002012  003454                  MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
014050  000137  014462                          JMP     PWCON           ;GO SERVICE POWER FAIL
014054                                  4$:     READEF  #EF.START       ;CHECK IF START
014054  012700  000040                          MOV     #EF.START,R0
014060  104447                                  TRAP    C$REFG
014062                                          BNCOMPLETE      RESTART ;NO - SKIP
014062  103034                                  BCC     RESTART

                                        ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
                                        ;       PASS COUNT, AND ERROR COUNT.

014064  013737  002012  003100                  MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
014072  005037  003444                  RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
014076  012700  003244                          MOV     #ERRCNT,R0
014102  012701  000100                          MOV     #64.,R1         ;GET A COUNT
014106  005020                          1$:     CLR     (R0)+           ;CLEAR AN ERROR COUNTER STORAGE AREA
014110  005301                                  DEC     R1
014112  001375                                  BNE     1$              ;LOOP TILL ALL CLEARED
014114  012737  003242  003242                  MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
014122  012737  177777  003446                  MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
014130  012737  177777  003014                  MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
014136  032737  040000  013722          LAB:    BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
014144  001002                                  BNE     5$              ;YES - SKIP
014146  005037  013724                          CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
014152  000432                          5$.     BR      SETDON
```

```
1254  014154                                  RESTART:
1255  014154                                          READEF  #EF.RESTART     ;CHECK IF RESTART
 (3)  014154  012700  000037                     MOV     #EF.RESTART,R0
 (3)  014160  104447                             TRAP    C$REFG
1256  014162                                          BCOMPLETE       RSTRT   ;NO - SKIP
 (2)  014162  103743                             BCS     RSTRT
1257  014164                                  CONTINUE:
1258  014164                                          READEF  #EF.CONTINUE    ;TEST IF CONTINUE
 (3)  014164  012700  000036                     MOV     #EF.CONTINUE,R0
 (3)  014170  104447                             TRAP    C$REFG
1259  014172                                          BCOMPLETE       PWCON
 (2)  014172  103533                             BCS     PWCON
1260                                          ;         ON CONTINUE PICK UP UNIT LAST UNDER TEST
1261  014174                                          READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
 (3)  014174  012700  000035                     MOV     #EF.NEW,R0
 (3)  014200  104447                             TRAP    C$REFG
1262  014202                                          BCOMPLETE       PASNEW
 (2)  014202  103403                             BCS     PASNEW
1263  014204                                  NXTPAS:
1264  014204  005737  003100                     TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
1265  014210  001013                             BNE     SETDON          ;NO - SKIP
1266  014212  005237  003444                 PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
1267  014216  012737  003242  003242             MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1268  014224  013737  002012  003100             MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
1269  014232  012737  177777  003446             MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
1270  014240  005237  003446                 SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
1271  014244  005337  003100                     DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
1272  014250  062737  000002  003242             ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
1273  014256  013700  003446                     MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
1274  014262  012702  003032                     MOV     #RLBAS,R2
1275  014266                                      GPHARD  R0,R1
 (3)  014266  104442                             TRAP    C$GPHRD
 (3)  014270  010001                             MOV     R0,R1
1276  014272                                      BCOMPLETE       7$      ;SKIP IF GOOD PARAM
 (2)  014272  103406                             BCS     7$
1277  014274  005737  003454                     TST     PWRFLG          ;RECENT POWER FAILURE
1278  014300  001741                             BEQ     NXTPAS  ;NO
1279  014302  005337  003454                     DEC     PWRFLG          ;ACCOUNT FOR DRIVE
1280  014306  000736                             BR      NXTPAS
1281  014310  012122                      7$:    MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
1282  014312  012122                             MOV     (R1)+,(R2)+     ;                   VECTOR
1283  014314  005721                             TST     (R1)+           ;BUMP PAST PRIORITY
1284  014316  012137  002302                     MOV     (R1)+,T.DRIVE
1285  014322  012122                             MOV     (R1)+,(R2)+
1286  014324  022737  000001  002302             CMP     #1,T.DRIVE
1287  014332  001426                             BEQ     65$
1288  014334  012737  000776  002312             MOV     #510.,NXTHL
1289  014342  012737  000777  002306             MOV     #511.,HLMTW
1290  014350  012737  001000  002314             MOV     #512.,GBND
1291  014356  012737  177600  002316             MOV     #177600,CAMSK
1292  014364  012737  177600  002320             MOV     #177600,DIRMSK
1293  014372  012737  177600  002322             MOV     #177600,HDCYL
1294  014400  012737  177000  002310             MOV     #177000,CLRBYT
1295  014406  000425                             BR      PWCON
1296
1297  014410  012737  000377  002306  65$:       MOV     #255.,HLMTW
```

```
1298  014416  012737  000400  002314          MOV     #256.,GBND
1299  014424  012737  077600  002316          MOV     #77600,CAMSK
1300  014432  012737  077600  002320          MOV     #77600,DIRMSK
1301  014440  012737  077600  002322          MOV     #77600,HDCYL
1302  014446  012737  000376  002312          MOV     #254.,NXTHL
1303  014454  012737  177400  002310          MOV     #177400,CLRBYT
1304
1305  014462  032737  020000  013722  PWCON:  BIT     #HICYL,MISWIW
1306  014470  001003                          BNE     1$
1307  014472  013737  002306  013726          MOV     HLMTW,HILIMW
1308  014500                          1$:     SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
 (7)  014500  012746  000340                  MOV     #340,-(SP)
 (6)  014504  012746  015752                  MOV     #INTHLR,-(SP)
 (5)  014510  013746  003034                  MOV     RLVEC,-(SP)
 (4)  014514  012746  000003                  MOV     #3,-(SP)
 (3)  014520  104437                          TRAP    C$SVEC
 (2)  014522  062706  000010                  ADD     #10,SP
1309  014526                                  SETPRI  #0              ;SET PRIORITY
 (3)  014526  012700  000000                  MOV     #0,R0
 (3)  014532  104441                          TRAP    C$SPRI
1310  014534  013702  003032                  MOV     RLBAS,R2        ;SET RL11 BASE ADDRESS POINTER
1321                            ;CHECK IF POWER FAILURE WAIT IS NEEDED
1322
1323  014540  005737  003454                  TST     PWRFLG          ;NEEDED???
1324  014544  001472                          BEQ     8$              ;NO, SKIP
1325
1326  014546  013705  003036                  MOV     RLDRV,R5        ;DRIVE SELECT
1327  014552  052705  000200                  BIS     #CRDYMSK,R5     ;SET CRDY
1328  014556  010562  000000                  MOV     R5,RLCS(R2)     ;SELECT DRIVE
1329  014562  012701  000170                  MOV     #120.,R1        ;INITIALIZE WAIT COUNT
1330  014566  032762  000001  000000  9$:     BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET?
1331  014574  001056                          BNE     8$              ;YES START TEST
1332
1333  014576                                  WAITMS  #10.            ;WAIT A SECOND
1334  014610  005301                          DEC     R1              ;SIXTY GONE BY
1335  014612  001365                          BNE     9$              ;NO
1336  014614                                  PRINTF  #FMT24,#NOPWR
 (8)  014614  012746  006166                  MOV     #NOPWR,-(SP)
 (7)  014620  012746  011776                  MOV     #FMT24,-(SP)
 (6)  014624  012746  000002                  MOV     #2,-(SP)
 (3)  014630  010600                          MOV     SP,R0
 (4)  014632  104417                          TRAP    C$PNTF
 (4)  014634  062706  000006                  ADD     #6,SP
1337  014640                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11)  014640  005046                          CLR     -(SP)
(11)  014642  153716  003037                  BISB    RLDRV+1,(SP)
(10)  014646  012746  006142                  MOV     #DRVNAM,-(SP)
 (9)  014652  013746  003032                  MOV     RLBAS,-(SP)
 (8)  014656  012746  006131                  MOV     #BASADD,-(SP)
 (7)  014662  012746  011172                  MOV     #FMT5,-(SP)
 (6)  014666  012746  000005                  MOV     #5,-(SP)
 (3)  014672  010600                          MOV     SP,R0
 (4)  014674  104417                          TRAP    C$PNTF
 (4)  014676  062706  000014                  ADD     #14,SP
1338  014702                                  PRINTF  #FMT3
 (7)  014702  012746  011156                  MOV     #FMT3,-(SP)
```

```
   (6)  014706  012746  000001                MOV     #1,-(SP)
   (3)  014712  010600                        MOV     SP,R0
   (4)  014714  104417                        TRAP    C$PNTF
   (4)  014716  062706  000004                ADD     #4,SP
  1339  014722                                DODU    PSETNM          ;DROP DRIVE
   (3)  014722  013700  003446                MOV     PSETNM,R0
   (3)  014726  104451                        TRAP    C$DODU
  1340  014730                                DOCLN
   (3)  014730  104444                        TRAP    C$DCLN
  1341  014732                          8$:
  1342
  1343  014732                          ENDINIT
   (3)  014732                          L10015:
   (3)  014732  104411                        TRAP    C$INIT
  1344  014734                          ENDMOD
  1345
```

```
1347
1348                                    .SBTTL  AUTO DROP SECTION
1349
1350                                    ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
1351                                    ;''ADR'' FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
1352                                    ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
1353                                    ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
1354                                    ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
1355                                    ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
1356
1357  014734                            BGNAUTO
1358  014734  005037  003452                   CLR     TRPFLG          ;CLEAR TRAP FLAG
1359  014740                            SETVEC  ERRVEC,#TRPHAN,#340     ;SET UP TRAP VECTOR TO DETECT
 (7)  014740  012746  000340                   MOV     #340,-(SP)
 (6)  014744  012746  015744                   MOV     #TRPHAN,-(SP)
 (5)  014750  013746  003234                   MOV     ERRVEC,-(SP)
 (4)  014754  012746  000003                   MOV     #3,-(SP)
 (3)  014760  104437                           TRAP    C$SVEC
 (2)  014762  062706  000010                   ADD     #10,SP
1360                                                                    ;/NON-EXISTENT CONTROLLER
1361  014766  013702  003032                   MOV     RLBAS,R2        ;GET RL11 BASE ADDRESS
1362  014772  005762  000000                   TST     RLCS(R2)        ;ACCESS DRIVE CONTROLLER ADDRESS
1363  014776  005737  003452                   TST     TRPFLG          ;DID TRAP OCCUR?
1364  015002  001447                           BEQ     1$              ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
1365  015004                                   PRINTF  #FMT24,#NOCTLR  ;ELSE, PRINT MSG. 'DRIVE DROPPED - NO CONTROLLER''
 (8)  015004  012746  007635                   MOV     #NOCTLR,-(SP)
 (7)  015010  012746  011776                   MOV     #FMT24,-(SP)
 (6)  015014  012746  000002                   MOV     #2,-(SP)
 (3)  015020  010600                           MOV     SP,R0
 (4)  015022  104417                           TRAP    C$PNTF
 (4)  015024  062706  000006                   ADD     #6,SP
1366  015030                                   PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 015030  005046                           CLR     -(SP)
 (11) 015032  153716  003037                   BISB    RLDRV+1,(SP)
 (10) 015036  012746  006142                   MOV     #DRVNAM,-(SP)
 (9)  015042  013746  003032                   MOV     RLBAS,-(SP)
 (8)  015046  012746  006131                   MOV     #BASADD,-(SP)
 (7)  015052  012746  011172                   MOV     #FMT5,-(SP)
 (6)  015056  012746  000005                   MOV     #5,-(SP)
 (3)  015062  010600                           MOV     SP,R0
 (4)  015064  104417                           TRAP    C$PNTF
 (4)  015066  062706  000014                   ADD     #14,SP
1367                                                                    ;PRINT DRIVE INFORMATION
1368  015072                                   PRINTF  #FMT3
 (7)  015072  012746  011156                   MOV     #FMT3,-(SP)
 (6)  015076  012746  000001                   MOV     #1,-(SP)
 (3)  015102  010600                           MOV     SP,R0
 (4)  015104  104417                           TRAP    C$PNTF
 (4)  015106  062706  000004                   ADD     #4,SP
1369
1370  015112                                   DODU    PSETNM          ;DO DROP UNIT ON DRIVE
 (3)  015112  013700  003446                   MOV     PSETNM,R0
 (3)  015116  104451                           TRAP    C$DODU
1371  015120  000460                           BR      2$              ;BRANCH TO EXIT
1372  015122  013705  003036           1$:     MOV     RLDRV,R5        ;ELSE, GET DRIVE NUMBER
1373  015126  052705  000200                   BIS     #CRDYMSK,R5     ;SET CONTROLLER READY
```

```
 1374  015132  010562  000000                MOV     R5,RLCS(R2)       ;LOAD IN THE DRIVE NUMBER
 1375  015136  032762  000001  000000        BIT     #DRDYMSK,RLCS(R2)         ;IS DRIVE READY?
 1376  015144  001046                        BNE     2$                ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
 1377  015146                                PRINTF  #FMT24,#NOTRDY    ;PRINT MSG. 'DRIVE DROPPED - DID NOT RESPOND
  (8)  015146  012746  007673                MOV     #NOTRDY,-(SP)
  (7)  015152  012746  011776                MOV     #FMT24,-(SP)
  (6)  015156  012746  000002                MOV     #2,-(SP)
  (3)  015162  010600                        MOV     SP,R0
  (4)  015164  104417                        TRAP    C$PNTF
  (4)  015166  062706  000006                ADD     #6,SP
 1378                                                                 ;/WITH 'READY'''
 1379  015172                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<8,RLDRV+1>
 (11)  015172  005046                        CLR     -(SP)
 (11)  015174  153716  003037                BISB    RLDRV+1,(SP)
 (10)  015200  012746  006142                MOV     #DRVNAM,-(SP)
  (9)  015204  013746  003032                MOV     RLBAS,-(SP)
  (8)  015210  012746  006131                MOV     #BASADD,-(SP)
  (7)  015214  012746  011172                MOV     #FMT5,-(SP)
  (6)  015220  012746  000005                MOV     #5,-(SP)
  (3)  015224  010600                        MOV     SP,R0
  (4)  015226  104417                        TRAP    C$PNTF
  (4)  015230  062706  000014                ADD     #14,SP
 1380                                                                 ;PRINT DRIVE INFORMATION
 1381  015234                                PRINTF  #FMT3
  (7)  015234  012746  011156                MOV     #FMT3,-(SP)
  (6)  015240  012746  000001                MOV     #1,-(SP)
  (3)  015244  010600                        MOV     SP,R0
  (4)  015246  104417                        TRAP    C$PNTF
  (4)  015250  062706  000004                ADD     #4,SP
 1382  015254                                DODU    PSETNM            ;DO DROP UNIT ON DRIVE
  (3)  015254  013700  003446                MOV     PSETNM,R0
  (3)  015260  104451                        TRAP    C$DODU
 1383  015262                        2$:     CLRVEC  ERRVEC            ;RELEASE ERROR VECTOR
  (3)  015262  013700  003234                MOV     ERRVEC,R0
  (3)  015266  104436                        TRAP    C$CVEC
 1384  015270                        ENDAUTO
  (3)  015270                        L10016:
  (3)  015270  104461                        TRAP    C$AUTO
 1385
```

```
1387
1388                                        .SBTTL  CLEANUP CODE SECTION
1389
1390  015272                                BGNMOD  CLNCODE
1391  015272                                BGNCLN
1392
1393  015272                                        SETVEC  ERRVEC,#TRPHAN,#340
 (7)  015272  012746  000340                         MOV     #340,-(SP)
 (6)  015276  012746  015744                         MOV     #TRPHAN,-(SP)
 (5)  015302  013746  003234                         MOV     ERRVEC,-(SP)
 (4)  015306  012746  000003                         MOV     #3,-(SP)
 (3)  015312  104437                                 TRAP    C$SVEC
 (2)  015314  062706  000010                         ADD     #10,SP
1394
1395  015320                                        SETPRI #7                       ;SET PRIORITY TO 7
 (3)  015320  012700  000007                         MOV     #7,R0
 (3)  015324  104441                                 TRAP    C$SPRI
1396  015326  032762  000200  000000  2$:           BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
1397  015334  001407                                BEQ     3$                      ;NO LOOP UNTIL READY
1398  015336  053762  003036  000000                BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
1399  015344  032762  000001  000000                BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
1400  015352  001005                                BNE     5$                      ;NO - SKIP
1401  015354                                3$:     WAITMS  #3                      ;WAIT 300 MS
1402  015366                                5$:     CLRVEC  RLVEC                   ;RELEASE VEC
 (3)  015366  013700  003034                         MOV     RLVEC,R0
 (3)  015372  104436                                 TRAP    C$CVEC
1403  015374  005737  003454                        TST     PWRFLG                  ;PWR FAIL SET
1404  015400  001402                                BEQ     7$                      ;NO
1405  015402  005337  003454                        DEC     PWRFLG
1406  015406                                7$:     CLRVEC  ERRVEC
 (3)  015406  013700  003234                         MOV     ERRVEC,R0
 (3)  015412  104436                                 TRAP    C$CVEC
1407  015414                                        BRESET                          ;TAKE CARE OF LSI-11
 (3)  015414  104433                                 TRAP    C$RESET
1408
1409  015416                                ENDCLN
 (3)  015416                                L10017:
 (3)  015416  104412                                 TRAP    C$CLEAN
1410
1411  015420                                BGNDU
1412  015420  000240                                NOP
1413  015422                                ENDDU
 (3)  015422                                L10020:
 (3)  015422  104453                                 TRAP    C$DU
1414
1415  015424                                ENDMOD
1416
```

```
                                        .SBTTL  GLOBAL SUBROUTINES

                                        BGNMOD  GLBSUB

015424  012737  000160  002116  TIME:   MOV     #160,L$DLY              ;GET OUTER DELAY LOOP
015432  005237  003466                  INC     TIM.US                  ;US-WAIT ROUTINE INDICATOR
015436  013737  003456  003462          MOV     XDELAY,MININC           ;SAVE ORIGINAL US-WAIT
015444  005437  003456                  NEG     XDELAY                  ;GET NEGATIVE OF FACTOR
015450                                  READBUS                         ;Q - BUS?
(3) 015450  104407                      TRAP    C$RDBU
015452                                  BCOMPLETE 2$                    ;BRANCH - IF YES
(2) 015452  103420                      BCS     2$
015454                          1$:     DELAY   #1.                     ;WAIT
(2) 015454  012727  000001              MOV     #1.,(PC)+
(2) 015460  000000                      .WORD   0
(2) 015462  013727  002116              MOV     L$DLY,(PC)+
(2) 015466  000000                      .WORD   0
(2) 015470  005367  177772              DEC     -6(PC)
(2) 015474  001375                      BNE     .-4
(2) 015476  005367  177756              DEC     -22(PC)
(2) 015502  001367                      BNE     .-20
015504  005237  003456                  INC     XDELAY                  ;WAIT FACTOR EXPIRED?
015510  002761                          BLT     1$                      ;BRANCH - IF NO
015512  000422                          BR      4$                      ;GET TIME
015514  012737  000065  002116  2$:     MOV     #65,L$DLY               ;GET OUTER DELAY LOOP
015522                          3$:     DELAY   #1.                     ;WAIT WITH RESPECT TO FONZ BUS
(2) 015522  012727  000001              MOV     #1.,(PC)+
(2) 015526  000000                      .WORD   0
(2) 015530  013727  002116              MOV     L$DLY,(PC)+
(2) 015534  000000                      .WORD   0
(2) 015536  005367  177772              DEC     -6(PC)
(2) 015542  001375                      BNE     .-4
(2) 015544  005367  177756              DEC     -22(PC)
(2) 015550  001367                      BNE     .-20
015552  005237  003456                  INC     XDELAY                  ;WAIT FACTOR EXPIRED?
015556  002761                          BLT     3$                      ;BRANCH - IF NO
015560  063737  003462  003122  4$:     ADD     MININC,TEMPO            ;GET TIME EXPIRED
015566  000207                          RTS     PC                      ;RETURN


015570  012737  000160  002116  XTIME:  MOV     #160,L$DLY              ;GET OUTER DELAY LOOP
015576  005037  003466                  CLR     TIM.US                  ;MS. WAIT INDICATOR
015602  013737  003460  003472          MOV     YDELAY,MAJINC           ;SAVE ORIGINAL WAIT MS
015610  006337  003460                  ASL     YDELAY                  ;MULTIPLY BY FACTOR 4
015614  006337  003460                  ASL     YDELAY                  ;---------
015620  005437  003460                  NEG     YDELAY                  ;GET NEGATIVE OF RESULT
015624                                  READBUS                         ;Q - BUS?
(3) 015624  104407                      TRAP    C$RDBU
015626                                  BNCOMPLETE      1$              ;BRANCH - IF NO
(2) 015626  103023                      BCC     1$
015630  012737  000150  002116          MOV     #150,L$DLY              ;GET OUTER DELAY LOOP
015636                          2$:     DELAY   #20                     ;WAIT WITH RESPECT TO FONZ BUS
(2) 015636  012727  000020              MOV     #20,(PC)+
(2) 015642  000000                      .WORD   0
(2) 015644  013727  002116              MOV     L$DLY,(PC)+
```

```
 (2)  015650  000000                          .WORD   0
 (2)  015652  005367  177772                  DEC     -6(PC)
 (2)  015656  001375                          BNE     .-4
 (2)  015660  005367  177756                  DEC     -22(PC)
 (2)  015664  001367                          BNE     .-20
1451  015666  005237  003460                  INC     YDELAY                  ;WAIT FACTOR EXPIRED
1452  015672  002761                          BLT     2$                      ;BRANCH - IF NO
1453  015674  000417                          BR      3$                      ;GET TIME
1454  015676                        1$:       DELAY   #10                     ;WAIT
 (2)  015676  012727  000010                  MOV     ##10,(PC)+
 (2)  015702  000000                          .WORD   0
 (2)  015704  013727  002116                  MOV     L$DLY,(PC)+
 (2)  015710  000000                          .WORD   0
 (2)  015712  005367  177772                  DEC     -6(PC)
 (2)  015716  001375                          BNE     .-4
 (2)  015720  005367  177756                  DEC     -22(PC)
 (2)  015724  001367                          BNE     .-20
1455  015726  005237  003460                  INC     YDELAY                  ;WAIT FACTOR EXPIRED?
1456  015732  002761                          BLT     1$                      ;BRANCH - IF NC
1457  015734  063737  003472  003464  3$:     ADD     MAJINC,TEMP             ;GET EXPIRED TIME
1458  015742  000207                          RTS     PC                      ;RETURN
1459
1460
1461
1462  015744                                BGNSRV
1463
1464                                        ;TRAP HANDLER INDICATES OCCURRENCE OF A TRAP.
1465
1466  015744  005237  003452                TRPHAN: INC     TRPFLG
1467
1468  015750                                ENDSRV
 (3)  015750                                L10021:
 (2)  015750  000002                                RTI
1469
1470  015752                                BGNSRV
1471
1472                                        ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.
1473
1474  015752                                INTHLR:
1475
1476  015752  012237  003050                        MOV     (R2)+,T.CS      ;STORE RL REGISTERS
1477  015756  012237  003052                        MOV     (R2)+,T.BA
1478  015762  012237  003054                        MOV     (R2)+,T.DA
1479  015766  011237  003056                        MOV     (R2),T.MP
1480  015772  012737  177777  003012              MOV     #-1,DONE        ;SET DONE FLAG
1481  016000  013702  003032                      MOV     RLBAS,R2        ;RESTORE R2
1482  016004                                      ABORTWAIT
1483
1484  016030                                ENDSRV
 (3)  016030                                L10022:
 (2)  016030  000002                                RTI
1485
```

```
1487
1488                                    ;       ERROR LIMIT CHECKING ROUTINE
1489                                    ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
1490
1491
1492  016032  027737  165204  013732  CKERLM: CMP     @ERRPOINT,ERLIMW        ;TEST IF ERROR LIMIT EXCEEDED
1493  016040  002453                          BLT     1$                      ;NO - SKIP
1494  016042                                  INLOOP                          ;CHECK IF IN ERROR LOOP
 (3)  016042  104420                          TRAP    C$INLP
1495  016044                                  .COMPLETE       1$              ;YES - SKIP
 (2)  016044  103451                          BCS     1$
1496  016046                                  PRINTF  #FMT25,ERLIMW,#MEXERS
 (9)  016046  012746  010657                  MOV     #MEXERS,-(SP)
 (8)  016052  013746  013732                  MOV     ERLIMW,-(SP)
 (7)  016056  012746  012003                  MOV     #FMT25,-(SP)
 (6)  016062  012746  000003                  MOV     #3,-(SP)
 (3)  016066  010600                          MOV     SP,R0
 (4)  016070  104417                          TRAP    C$PNTF
 (4)  016072  062706  000010                  ADD     #10,SP
1497  016076                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 016076  005046                          CLR     -(SP)
 (11) 016100  153716  003037                  BISB    RLDRV+1,(SP)
 (10) 016104  012746  006142                  MOV     #DRVNAM,-(SP)
 (9)  016110  013746  003032                  MOV     RLBAS,-(SP)
 (8)  016114  012746  006131                  MOV     #BASADD,-(SP)
 (7)  016120  012746  011172                  MOV     #FMT5,-(SP)
 (6)  016124  012746  000005                  MOV     #5,-(SP)
 (3)  016130  010600                          MOV     SP,R0
 (4)  016132  104417                          TRAP    C$PNTF
 (4)  016134  062706  000014                  ADD     #14,SP
1498  016140                                  PRINTF  #FMT3
 (7)  016140  012746  011156                  MOV     #FMT3,-(SP)
 (6)  016144  012746  000001                  MOV     #1,-(SP)
 (3)  016150  010600                          MOV     SP,R0
 (4)  016152  104417                          TRAP    C$PNTF
 (4)  016154  062706  000004                  ADD     #4,SP
1499  016160                                  DODU    PSETNM                  ;DROP DRIVE
 (3)  016160  013700  003446                  MOV     PSETNM,R0
 (3)  016164  104451                          TRAP    C$DODU
1500  016166                                  DOCLN                           ;GO TO CLEAN UP
 (3)  016166  104444                          TRAP    C$DCLN
1501  016170  000207                  1$:     RTS     PC
1502
1503                                    ;       READ AND STORE ALL RL11 REGISTERS
1504  016172  016237  000000  003050  READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
1505  016200  016237  000002  003052          MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
1506  016206  016237  000004  003054          MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
1507  016214  016237  000006  003056          MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
1508  016222  000207                          RTS     PC              ;RETURN
1509
1510                                    ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
1511  016224  011646                  WAITIN: MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
1512  016226  005066  000002                  CLR     2(SP)           ;CLEAR FOR POINTER
1513  016232  032762  000200  000000          BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
1514  016240  001420                          BEQ     4$              ;NO - SKIP TO WAIT
1515  016242  004737  016172                  JSR     PC,READRL       ;READ ALL RL REGS
```

```
1516  016246  005737  003012                TST     DONE            ;TEST IF INTERRUPT OCCURRED
1517  016252  001435                        BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
1518  016254  012766  006306  000002  1$:   MOV     #MTOSLOW,2(SP)  ;ELSE SET TOO SLOW ERROR POINTER
1519  016262  032737  002000  003050        BIT     #OPIERR,T.CS    ;TEST IF OPI SET
1520  016270  001403                        BEQ     2$              ;NO - SKIP
1521  016272  012766  006326  000002        MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
1522  016300  000207              2$:       RTS     PC              ;RETURN
1523  016302                      4$:       WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
1524  016314  032762  000200  000000        BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
1525  016322  001006                        BNE     3$              ;YES - SKIP
1526  016324  004737  016172                JSR     PC,READRL       ;READ RL REGS
1527  016330  012766  006377  000002        MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
1528  016336  000760                        BR      2$              ;SKIP
1529  016340  005737  003012          3$:   TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
1530  016344  001343                        BNE     1$              ;YES - SKIP TO SET TOO SLOW
1531  016346  004737  016172          5$:   JSR     PC,READRL       ;READ RL REGS
1532  016352  012766  006344  000002        MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
1533  016360  000747                        BR      2$              ;GO TO RETURN
1534
1535                              ;         OPERATION AND TEST INITIALIZE ROUTINE
1536  016362  005037  003010          TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
1537  016366  105037  003451                  CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
1538  016372  005037  003020                  CLR     MORECE          ;RESET MORE COMPARE ERRORS
1539  016376  000207                        RTS     PC
1540
1541                              ;         GET STATUS AND GET STATUS WITH RESET ROUTINE
1542  016400  013746  003132          GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
1543  016404  012737  000013  003132        MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
1544  016412  000412                        BR      GSTATG
1545  016414  013746  003132          GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
1546  016420  012737  000003  003132        MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
1547  016426  000404                        BR      GSTATG
1548  016430  013746  003132          GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
1549  016434  005037  003132                  CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
1550  016440  010346                GSTATG: MOV     R3,-(SP)        ;STORE R3
1551  016442  013703  003006                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1552  016446  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1553  016450  016663  000004  002410        MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
1554  016456  162763  000004  002410        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1555  016464  010337  003006                MOV     R3,SSINDX       ;STORE IT BACK
1556  016470  010046                        MOV     R0,-(SP)        ;STORE R0
1557  016472  010146                        MOV     R1,-(SP)        ;STORE R1
1558  016474  012737  000002  003022        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1559  016502  032737  000010  003132        BIT     #DRSET,TEMP4    ;TEST IF DRIVE RESET
1560  016510  001460                        BEQ     11$             ;NO - SKIP
1561  016512  032762  040000  000000        BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
1562  016520  001405                        BEQ     49$             ;NO - SKIP
1563  016522                                WAITMS  #3              ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
1564  016534  012701  000062          49$:  MOV     #50.,R1         ;INITIALIZE WAIT COUNT
1565  016540  004737  016430          50$:  JSR     PC,GSTAT        ;GET DRIVE STATUS
1566  016544  017230                        3$
1567  016546  032737  000001  003050        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1568  016554  001054                        BNE     5$              ;YES - GO DO CLEAR
1569  016556  032737  000020  003056        BIT     #HOSTAT,T.MP    ;ELSE TEST IF HEADS OUT
1570  016564  001010                        BNE     51$             ;YES - BYPASS RELOAD WAIT FLAG SETTING
1571  016566  032737  144000  003056        BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
```

```
1572                                                                    ;THAT CAUSED HEADS TO
1573                                                                    ;UNLOAD
1574  016574  001444                    BEQ     5$              ;NO - SKIP
1575  016576  052737  040000  003010    BIS     #RELDWT,OPFLAG  ;ELSE SET WAIT FLAG
1576  016604  000440                    BR      5$              ;SKIP TO CLEAR
1577  016606  032737  040000  003050  51$:  BIT  #DRVERR,T.CS    ;TEST IF DRIVE ERROR NOW
1578  016614  001034                    BNE     5$              ;YES - SKIP TO CLEAR
1579  016616                            WAITMS  #1              ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HEADS OUT
1580  016630  005301                    DEC     R1              ;DEC WAIT COUNTER
1581  016632  001342                    BNE     50$             ;IF NOT DONE, LOOP
1582  016634  012703  010541            MOV     #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
1583  016640                            ERRHRD  10001.,,ERR1
 (4)  016640  104456                    TRAP    C$ERHRD
 (5)  016642  023421                    .WORD   10001
 (5)  016644  000000                    .WORD   0
 (5)  016646  012070                    .WORD   ERR1
1584  016650  000565                    BR      14$             ;EXIT
1585  016652  005737  003132  11$:      TST     TEMP4           ;TEST IF SAVE REGISTERS
1586  016656  001013                    BNE     5$              ;NO SKIP
1587  016660  012701  000004            MOV     #4,R1           ;SET SAVE COUNT
1588  016664  012703  003050            MOV     #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
1589  016670  014346            8$:     MOV     -(R3),-(SP)     ;PUT REG ON STACK
1590  016672  005301                    DEC     R1              ;DEC COUNT
1591  016674  001375                    BNE     8$              ;LOOP UNTIL ALL SAVED
1592  016676  012737  000003  003044    MOV     #GETSTAT,L.DA   ;SET FOR GET STATUS
1593  016704  000403                    BR      6$              ;SKIP
1594  016706  013737  003132  003044  5$:  MOV  TEMP4,L.DA      ;INSERT PRESET FOR STATUS
1595  016714                          6$:
1596  016714  005037  003012            CLR     DONE            ;CLEAR INTERRUPT FLAG
1597  016720  013737  003036  003040    MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
1598  016726  042737  002000  003040    BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1599  016734  052737  000104  003040    BIS     #GTSTAT,L.CS
1600  016742  013762  003044  000004    MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
1601  016750  013762  003040  000000    MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
1602  016756                            WAITUS  #1              ;WAIT 100 US FOR INTERRUPT
1603  016770  005737  003012            TST     DONE            ;CHECK IF INTERRUPT OCCURRED
1604  016774  001504                    BEQ     1$              ;NO - SKIP
1605  016776  013737  003056  003064  4$:  MOV  T.MP,T.STAT     ;STORE MP REGISTER
1606  017004  042737  177770  003064    BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
1607  017012  032737  000010  003044    BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
1608  017020  001503                    BEQ     3$              ;NO - SKIP TO EXIT
1609  017022  032737  040000  003010    BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
1610  017030  001427                    BEQ     12$             ;NO - SKIP
1611  017032  012701  001130            MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
1612  017036  032762  000001  000000  13$:  BIT  #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
1613  017044  001021                    BNE     12$             ;YES - SKIP
1614  017046                            WAITMS  #1              ;CALL WAIT
1615  017060  005301                    DEC     R1              ;DEC COUNT
1616  017062  001365                    BNE     13$             ;LOOP IF NOT 0
1617  017064  004737  016430            JSR     PC,GSTAT        ;GET DRIVE STATUS
1618  017070  017230                    3$                      ;ERROR RETURN
1619  017072  012703  010606            MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
1620  017076                            ERRHRD  10003.,,ERR1
 (4)  017076  104456                    TRAP    C$ERHRD
 (5)  017100  023423                    .WORD   10003
 (5)  017102  000000                    .WORD   0
```

```
  (5)  017104  012070                          .WORD   ERR1
1621   017106  000446                          BR      14$             ;GO TO EXIT
1622   017110                          12$:    WAITUS  #10.            ;WAIT FOR 1MS
1623   017122  004737  016430                  JSR     PC,GSTAT        ;GET DRIVE STATUS
1624   017126  017230                          3$
1625   017130  032737  100000  003050          BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
1626   017136  001434                          BEQ     3$              ;NO - SKIP
1627   017140  032737  001000  003056          BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
1628   017146  001403                          BEQ     7$              ;YES SKIP
1629   017150  012703  006433                  MOV     #VCNRST,R3      ;SET REASON POINTER
1630   017154  000417                          BR      2$              ;EXIT
1631   017156  032737  040000  003050  7$:     BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
1632   017164  001405                          BEQ     9$              ;NO - SKIP
1633   017166                                  ERRHRD  10004.,,ERR6
  (4)  017166  104456                          TRAP    C$ERHRD
  (5)  017170  023424                          .WORD   10004
  (5)  017172  000000                          .WORD   0
  (5)  017174  012372                          .WORD   ERR6
1634   017176  000412                          BR      14$             ;EXIT
1635   017200  012703  006454          9$:     MOV     #UNXERR,R3      ;SET REASON POINTER
1636   017204  000403                          BR      2$              ;EXIT
1637   017206  004737  016224          1$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
1638   017212  012603                          MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
1639   017214                          2$:     ERRHRD  10002.,,ERR1
  (4)  017214  104456                          TRAP    C$ERHRD
  (5)  017216  023422                          .WORD   10002
  (5)  017220  000000                          .WORD   0
  (5)  017222  012070                          .WORD   ERR1
1640   017224  005037  003022          14$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1641   017230  005737  003132          3$:     TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
1642   017234  001007                          BNE     22$             ;NO - SKIP
1643   017236  012703  003040                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE
1644   017242  012701  000004                  MOV     #4,R1           ;SET REGISTER COUNT
1645   017246  012623                  20$:    MOV     (SP)+,(R3)+     ;RESTORE REG
1646   017250  005301                          DEC     R1              ;DEC COUNT
1647   017252  001375                          BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
1648   017254  162737  000002  003006  22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1649   017262  012601                          MOV     (SP)+,R1        ;RESTORE R1
1650   017264  012600                          MOV     (SP)+,R0        ;RESTORE R0
1651   017266  012603                          MOV     (SP)+,R3        ;RESTORE R3
1652   017270  012637  003132                  MOV     (SP)+,TEMP4     ;RESTORE TEMP4
1653   017274  005737  003022                  TST     ERRSWI          ;TEST IF ERROR RETURN
1654   017300  001403                          BEQ     99$             ;YES - SKIP
1655   017302  063716  003022                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1656   017306  000207                          RTS     PC
1657   017310  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1658   017314  000207                          RTS     PC
1659
1660
1661                                    ;       SEEK ROUTINE
1662   017316  012737  177777  003124  XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
1663   017324  000402                          BR      XSEEK1
1664   017326  005037  003124          XSEEK:  CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
1665   017332  010346                  XSEEK1: MOV     R3,-(SP)        ;STORE R3
1666   017334  013703  003006                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1667   017340  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
```

```
1668  017342  016663  000002  002410          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1669  017350  162763  000004  002410          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1670  017356  010337  003006                  MOV     R3,SSINDX       ;STORE IT BACK
1671  017362  010046                          MOV     R0,-(SP)
1672  017364  010146                          MOV     R1,-(SP)
1673  017366  010546                          MOV     R5,-(SP)        ;STORE REG
1674  017370  012737  000002  003022          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1675  017376  005037  003102                  CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
1676                                                                  ;    PAST GUARD BAND)
1677  017402  004737  022506                  JSR     PC,GETPOS       ;GET PRESENT POSITION
1678  017406  020040                          65$
1679  017410  013737  003110  003104          MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
1680  017416  023737  003106  002306          CMP     NEWCYL,HLMTW    ;TEST IF NEW IS GREATER THAN 255
1681  017424  003427                          BLE     3$              ;NO - SKIP
1682  017426  163737  002306  003106          SUB     HLMTW,NEWCYL    ;ELSE SUBTRACT 255.
1683  017434  013737  003106  003102          MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
1684  017442  013737  002306  003106          MOV     HLMTW,NEWCYL    ;SET NEWCYL AS 255.
1685  017450  022737  000001  002302          CMP     #1,T.DRIVE
1686  017456  001424                          BEQ     6$
1687  017460  162737  000001  003106          SUB     #1,NEWCYL
1688  017466  012737  000001  003114          MOV     #1,DESSGN
1689  017474  012737  000001  003112          MOV     #1,DESDIF
1690  017502  000451                          BR      18$
1691  017504  005737  003106          3$:     TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
1692  017510  100007                          BPL     6$              ;NO - SKIP
1693  017512  005437  003106                  NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
1694  017516  013737  003106  003102          MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
1695  017524  005037  003106                  CLR     NEWCYL          ;AND SET NEWCYL TO 0
1696  017530  013705  003110          6$:     MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
1697  017534  163705  003106                  SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
1698  017540  100005                          BPL     13$             ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
1699  017542  012737  000001  003114          MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
1700  017550  005405                          NEG     R5              ;MAKE DIFFERENCE POSITIVE
1701  017552  000402                          BR      14$             ;SKIP
1702  017554  005037  003114          13$:    CLR     DESSGN          ;SET SIGN FOR REVERSE
1703  017560  010537  003112          14$:    MOV     R5,DESDIF       ;STORE DIFFERENCE
1704  017564  005737  003102                  TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
1705  017570  001416                          BEQ     18$             ;NO - SKIP
1706  017572  023737  003106  00230(          CMP     NEWCYL,HLMTW    ;CHECK IF NEW CYL IS 255.
1707  017600  001007                          BNE     17$             ;NO - SKIP
1708  017602  012737  000001  003114          MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
1709                                                                  ;(INNER GUARD BAND)
1710  017610  022737  000001  002302          CMP     #1,T.DRIVE
1711  017616  001003                          BNE     18$
1712  017620  063737  003102  003112  17$:    ADD     DIFAUG,DESDIF
1713  017626                          18$:
1714  017626  012705  003040                  MOV     #L.CS,R5        ;GET L REG ADDRESS
1715  017632  012715  000106                  MOV     #SEEK,(R5)      ;SET FOR SEEK
1716  017636  053715  003036                  BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
1717  017642  042725  002000                  BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
1718  017646  005025                          CLR     (R5)+           ;CLEAR BUS ADDRESS
1719  017650  013715  003112                  MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
1720  017654  012700  000007                  MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
1721  017660  006315                  21$:    ASL     (R5)
1722  017662  005300                          DEC     R0
1723  017664  001375                          BNE     21$             ;LOOP UNTIL ALIGNED
```

```
 1724  017666  005737  003114                 TST     DESSGN          ;TEST SIGN
 1725  017672  001402                         BEQ     23$             ;SKIP IF 0
 1726  017674  052715  000004                 BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
 1727  017700  005737  003116         23$:    TST     DESHD           ;TEST IF HEAD 0
 1728  017704  001402                         BEQ     25$             ;YES - SKIP
 1729  017706  052715  000020                 BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
 1730  017712  052725  000001         25$:    BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
 1731  017716  004737  020444                 JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
 1732  017722  020040                         65$
 1733  017724  005037  003012                 CLR     DONE            ;CLEAR INTERRUPT FLAG
 1734  017730  005737  003124                 TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
 1735  017734  001041                         BNE     65$             ;YES - SKIP, DO NOT START SEEK
 1736  017736  014562  000004                 MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
 1737  017742  014562  000002                 MOV     -(R5),RLBA(R2)
 1738  017746  014562  000000                 MOV     -(R5),RLCS(R2)
 1739  017752                         30$:    WAITUS  #10.
 1740  017764  005737  003012                 TST     DONE            ;TEST IF INTERRUPT DONE
 1741  017770  001012                         BNE     32$             ;YES - SKIP
 1742  017772  004737  016224                 JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
 1743  017776  012603                         MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 1744  020000                                 ERRHRD  10005.,,ERR1
  (4)  020000  104456                         TRAP    C$ERHRD
  (5)  020002  023425                         .WORD   10005
  (5)  020004  000000                         .WORD   0
  (5)  020006  012070                         .WORD   ERR1
 1745  020010  005037  003022                 CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 1746  020014  000411                         BR      65$
 1747  020016  005737  003050         32$:    TST     T.CS            ;TEST IF ANY ERROR
 1748  020022  100006                         BPL     65$             ;NO - SKIP
 1749  020024                                 ERRHRD  10006.,,ERR6
  (4)  020024  104456                         TRAP    C$ERHRD
  (5)  020026  023426                         .WORD   10006
  (5)  020030  000000                         .WORD   0
  (5)  020032  012372                         .WORD   ERR6
 1750  020034  005037  003022                 CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 1751  020040  162737  000002  003006  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1752  020046  012605                         MOV     (SP)+,R5        ;RESTORE REGISTERS
 1753  020050  012601                         MOV     (SP)+,R1
 1754  020052  012600                         MOV     (SP)+,R0
 1755  020054  012603                         MOV     (SP)+,R3
 1756  020056  005737  003022                 TST     ERRSWI          ;TEST IF ERROR RETURN
 1757  020062  001403                         BEQ     99$             ;YES - SKIP
 1758  020064  063716  003022                 ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1759  020070  000207                         RTS     PC
 1760  020072  017616  000000         99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1761  020076  000207                         RTS     PC
 1762
 1819
 1821                                 ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
 1822                                 ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
 1823  020100  010346                 POSHDS: MOV     R3,-(SP)        ;SAVE REGS
 1824  020102  013703  003006                 MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1825  020106  005723                         TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1826  020110  016663  000002  002410         MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1827  020116  162763  000004  002410         SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1828  020124  010337  003006                 MOV     R3,SSINDX       ;STORE IT BACK
```

```
 1829  020130  010346                          MOV     R3,-(SP)
 1830  020132  010446                          MOV     R4,-(SP)
 1831  020134  012737  000002  003022          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1832  020142  004737  022506                  JSR     PC,GETPOS       ;GET CURRENT POSITION
 1833  020146  020406                          PH65$
 1834  020150  012704  000012                  MOV     #10.,R4         ;SET RETRY COUNT
 1835  020154                          BGNSEG
  (3)  020154  104404                          TRAP    C$BSEG
 1836  020156                          1$:     INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  020156  104420                          TRAP    C$INLP
 1837  020160                                  BNCOMPLETE      5$      ;NO - SKIP
  (2)  020160  103012                          BCC     5$
 1838  020162  004737  022506                  JSR     PC,GETPOS       ;ELSE GET POSITION
 1839  020166  020404                          60$
 1840  020170  023737  003110  003106          CMP     CURCYL,NEWCYL   ;CHECK IF AT INTENDED POSITION
 1841  020176  001017                          BNE     8$              ;NO - SKIP
 1842  020200  004737  021004                  JSR     PC,ONSWAP       ;SWAP OLDCYL AND NEWCYL
 1843  020204  000414                          BR      8$              ;SKIP
 1844  020206  013737  003110  003104  5$:     MOV     CURCYL,OLDCYL   ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
 1845  020214  023705  003110                  CMP     CURCYL,R5       ;CHECK IF HDS AT FINAL POSITION
 1846  020220  001471                          BEQ     60$             ;YES - GO TO EXIT
 1847  020222  003003                          BGT     7$              ;IF CURCYL > FINAL POSITION - SKIP
 1848  020224  005237  003106                  INC     NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
 1849  020230  000402                          BR      8$              ;SKIP
 1850  020232  005337  003106          7$:     DEC     NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
 1851  020236  004737  017326          8$:     JSR     PC,XSEEK        ;DO SEEK
 1852  020242  020404                          60$
 1853  020244  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT 300 MS
 1854  020250  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
 1855  020254  020404                          60$
 1856  020256  005737  003050                  TST     T.CS            ;TEST IF ANY ERROR
 1857  020262  100007                          BPL     10$             ;NO - SKIP
 1858  020264                                  ERRHRD  10008.,,ERR6
  (4)  020264  104456                          TRAP    C$ERHRD
  (5)  020266  023430                          .WORD   10008
  (5)  020270  000000                          .WORD   0
  (5)  020272  012372                          .WORD   ERR6
 1859  020274  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1860  020300  000441                          BR      60$
 1861  020302  004737  022506          10$:    JSR     PC,GETPOS       ;GET POSITION
 1862  020306  020404                          60$
 1863  020310  023737  003110  003106          CMP     CURCYL,NEWCYL   ;CHECK IF ARRIVED AT DESIRED PLACE
 1864  020316  001003                          BNE     15$             ;NO - SKIP
 1865  020320  012704  000012          14$:    MOV     #10.,R4         ;ELSE INIT RETRY COUNT
 1866  020324  000714                          BR      1$              ;GO DO NEXT SEEK
 1867  020326  005737  003114          15$:    TST     DESSGN          ;TEST IF GOING IN
 1868  020332  001017                          BNE     17$             ;YES - SKIP
 1869  020334  023737  003110  003106          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS DID NOT MOVE IN
 1870  020342  003366                          BGT     14$             ;YES - SKIP
 1871  020344  005304                  16$:    DEC     R4              ;DEC RETRY COUNT
 1872  020346  001333                          BNE     8$              ;DO ANOTHER SEEK IF NOT 0
 1873  020350  012703  007313                  MOV     #HDMOVF,R3      ;ELSE SET RESULT MESSAGE POINTER
 1874  020354                                  ERRHRD  10009.,,ERR1
  (4)  020354  104456                          TRAP    C$ERHRD
  (5)  020356  023431                          .WORD   10009
  (5)  020360  000000                          .WORD   0
```

```
 (5)  020362  012070                          .WORD   ERR1
1875  020364  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1876  020370  000405                          BR      60$
1877  020372  023737  003110  003106  17$:    CMP     CURCYL,NEWCYL   ;HDS SHOULD MOVE OUT, CHK THEY DID
1878  020400  002747                          BLT     14$             ;YES - SKIP
1879  020402  000760                          BR      16$             ;ELSE GO DEC AND RETRY
1880  020404                        20$:
1881  020404                        60$:
1882  020404                        ENDSEG
 (3)  020404                        10000$:
 (3)  020404  104405                          TRAP    C$ESEG
1883  020406  162737  000002  003006  PH65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1884  020414  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
1885  020416  012600                          MOV     (SP)+,R0
1886  020420  012603                          MOV     (SP)+,R3
1887  020422  005737  003022                  TST     ERRSWI          ;TEST IF ERROR RETURN
1888  020426  001403                          BEQ     99$             ;YES - SKIP
1889  020430  063716  003022                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1890  020434  000207                          RTS     PC
1891  020436  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1892  020442  000207                          RTS     PC
1893
1895                                ;       DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
1896                                ;       500MS FOR READY TO SET.
1897  020444  010346                RDYCHK: MOV     R3,-(SP)        ;STORE REGS
1898  020446  013703  003006                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1899  020452  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1900  020454  016663  000002  002410          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1901  020462  162763  000004  002410          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1902  020470  010337  003006                  MOV     R3,SSINDX       ;STORE IT BACK
1903  020474  010046                          MOV     R0,-(SP)
1904  020476  010146                          MOV     R1,-(SP)
1905  020500  010446                          MOV     R4,-(SP)
1906  020502  012737  000002  003022          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1907  020510  012701  011610                  MOV     #5000.,R1       ;SET WAIT COUNT
1908  020514  004737  016430          1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
1909  020520  020654                          4$
1910  020522  032737  000001  003050          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1911  020530  001053                          BNE     5$              ;YES - EXIT
1912  020532                                  WAITUS  #1
1913  020544  005301                          DEC     R1              ;DEC WAIT COUNT
1914  020546  001362                          BNE     1$              ;LOOP IF NOT 0
1915  020550  012703  010124                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
1916  020554  012704  011011                  MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
1917  020560                                  ERRHRD  10010.,,ERR5
 (4)  020560  104456                          TRAP    C$ERHRD
 (5)  020562  023432                          .WORD   10010
 (5)  020564  000000                          .WORD   0
 (5)  020566  012322                          .WORD   ERR5
1918  020570  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR      ONDS
1919  020574  004737  016430          2$:     JSR     PC,GSTAT        ;GET DRIVE STATU
1920  020600  020654                          4$
1921  020602  032737  000001  003050          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE REA
1922  020610  001007                          BNE     3$              ;YES - SKIP
1923  020612                                  WAITMS  #1              ;WAIT FOR 100MS
1924  020624  005301                          DEC     R1              ;DEC WAIT COUNTER
```

```
1925  020626  001362                          BNE     2$              ;LOOP UNTIL TIME DONE
1926  020630  032737  100000  003050  3$:     BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
1927  020636  001406                          BEQ     4$              ;NO - SKIP
1928  020640                                  ERRHRD  10011.,,ERR6    ;REPORT ALL ERRORS
 (4)  020640  104456                          TRAP    C$ERHRD
 (5)  020642  023433                          .WORD   10011
 (5)  020644  000000                          .WORD   0
 (5)  020646  012372                          .WORD   ERR6
1929  020650  005337  003244                  DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
1930  020654  005037  003022          4$:     CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1931  020660  162737  000002  003006  5$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1932  020666  012604                          MOV     (SP)+,R4        ;RESTORE REGS
1933  020670  012601                          MOV     (SP)+,R1
1934  020672  012600                          MOV     (SP)+,R0
1935  020674  012603                          MOV     (SP)+,R3
1936  020676  005737  003022                  TST     ERRSWI          ;TEST IF ERROR RETURN
1937  020702  001403                          BEQ     99$             ;YES - SKIP
1938  020704  063716  003022                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1939  020710  000207                          RTS     PC
1940  020712  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1941  020716  000207                          RTS     PC
1942
1943                                  ;       CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1944                                  ;       SELECTED BY SOFTWARE PARAMETER.
1945  020720  005037  003116          CHOSHD: CLR     DESHD           ;CLEAR TO HEAD 0
1946  020724  032737  010000  013722          BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1947  020732  001403                          BEQ     1$              ;NO - SKIP
1948  020734  013737  013730  003116          MOV     HEADW,DESHD     ;INSERT SPECIFIED HEAD
1949  020742  000207                  1$:     RTS     PC
1950
1951                                  ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1952                                  ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1953  020744  032737  010000  013722  SWAPHD: BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1954  020752  001011                          BNE     2$              ;YES - TAKE ABORT EXIT
1955  020754  005737  003116                  TST     DESHD           ;TEST IF HEAD ONE USED
1956  020760  001006                          BNE     2$              ;YES - TAKE ABORT EXIT
1957  020762  012737  000001  003116          MOV     #1,DESHD        ;ELSE SET FOR HEAD ONE
1958  020770  062716  000002                  ADD     #2,(SP)         ;BUMP PAST ABORT RETURN
1959  020774  000207                          RTS     PC              ;RETURN
1960  020776  017616  000000          2$:     MOV     @(SP),(SP)      ;GET ABORT DESTINATION
1961  021002  000207                  3$:     RTS     PC
1962
1963                                  ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1964  021004  010046                  ONSWAP: MOV     R0,-(SP)        ;STORE R0
1965  021006  013700  003104                  MOV     OLDCYL,R0       ;MOVE OLD TO R0
1966  021012  013737  0031C6  003104          MOV     NEWCYL,OLDCYL   ;MOVE NEW TO OLD
1967  021020  010037  003106                  MOV     R0,NEWCYL       ;PUT OLD IN NEW
1968  021024  012600                          MOV     (SP)+,R0        ;RESTORE R0
1969  021026  000207                          RTS     PC
1970
1972                                  ;       BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
1973                                  ;       FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
1974                                  ;       FILES TO LOOK LIKE ALL SECTORS OK.
1975  021030  005737  003500          CKBSVD: TST     BSFVAL          ;TEST IF BAD SECTORS STORED
1976  021034  001051                          BNE     5$              ;YES - EXIT
1977  021036                                  PRINTF  #FMT9,#BSNSTR   ;REPORT
```

```
  (8)  021036  012746  007540                    MOV     #BSNSTR,-(SP)
  (7)  021042  012746  011356                    MOV     #FMT9,-(SP)
  (6)  021046  012746  000002                    MOV     #2,-(SP)
  (3)  021052  010600                            MOV     SP,R0
  (4)  021054  104417                            TRAP    C$PNTF
  (4)  021056  062706  000006                    ADD     #6,SP
 1978  021062                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11) 021062  005046                            CLR     -(SP)
  (11) 021064  153716  003037                    BISB    RLDRV+1,(SP)
  (10) 021070  012746  006142                    MOV     #DRVNAM,-(SP)
  (9)  021074  013746  003032                    MOV     RLBAS,-(SP)
  (8)  021100  012746  006131                    MOV     #BASADD,-(SP)
  (7)  021104  012746  011172                    MOV     #FMT5,-(SP)
  (6)  021110  012746  '000005                   MOV     #5,-(SP)
  (3)  021114  010600                            MOV     SP,R0
  (4)  021116  104417                            TRAP    C$PNTF
  (4)  021120  062706  000014                    ADD     #14,SP
 1979  021124                                    PRINTF  #FMT3
  (7)  021124  012746  011156                    MOV     #FMT3,-(SP)
  (6)  021130  012746  000001                    MOV     #1,-(SP)
  (3)  021134  010600                            MOV     SP,R0
  (4)  021136  104417                            TRAP    C$PNTF
  (4)  021140  062706  000004                    ADD     #4,SP
 1980  021144  012737  177777  003502            MOV     #-1,SBSFIL      ;FORCE FILES TO NO ENTRIES
 1981  021152  012737  177777  003676            MOV     #-1,FBSFIL
 1982  021160  000207                  5$:       RTS     PC
 1983
 1985                                            ;       READ HEADERS ROUTINE.
 1986  021162  012737  000001  003132  XRDHDC:   MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
 1987  021170  000402                            BR      XRDHDG          ;GO DO IT
 1988  021172  005037  003132          XRDHD:    CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
 1989  021176  010346                  XRDHDG:   MOV     R3,-(SP)        ;STORE REGISTERS
 1990  021200  013703  003006                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1991  021204  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1992  021206  016663  000002  002410            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1993  021214  162763  000004  002410            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1994  021222  010337  003006                    MOV     R3,SSINDX       ;STORE IT BACK
 1995  021226  010046                            MOV     R0,-(SP)
 1996  021230  010146                            MOV     R1,-(SP)
 1997  021232  010446                            MOV     R4,-(SP)
 1998  021234  012737  000002  003022            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1999  021242  005737  003132                    TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
 2000  021246  001007                            BNE     2$              ;NO - SKIP
 2001  021250  012703  003050                    MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
 2002  021254  012701  000004                    MOV     #4,R1           ;SET COUNT
 2003  021260  014346                  1$:       MOV     -(R3),-(SP)     ;SAVE REGISTER
 2004  021262  005301                            DEC     R1              ;DEC COUNT
 2005  021264  001375                            BNE     1$              ;LOOP UNTIL ALL ARE SAVED
 2006  021266  004737  020444          2$:       JSR     PC,RDYCHK       ;CHECK DRIVE READY
 2007  021272  021542                            65$
 2008  021274  005037  003012                    CLR     DONE            ;CLEAR INTERRUPT FLAG
 2009  021300  012701  003040                    MOV     #L.CS,R1        ;GET ADDRESS OF LOAD REGS
 2010  021304  013711  003036                    MOV     RLDRV,(R1)      ;LOAD DRIVE NUMBER
 2011  021310  042711  002000                    BIC     #BIT10,(R1)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 2012  021314  052721  000110                    BIS     #RDHEAD,(R1)+   ;INSERT COMMAND
 2013  021320  005021                            CLR     (R1)+           ;CLEAR BA
```

```
2014  021322  005021                          CLR     (R1)+           ;CLEAR DA
2015  021324  014162  000004                  MOV     -(R1),RLDA(R2)  ;LOAD RL11 REGS
2016  021330  014162  000002                  MOV     -(R1),RLBA(R2)
2017  021334  014162  000000                  MOV     -(R1),RLCSR(R2)
2018  021340                          3$:     WAITUS  #10.            ;WAIT 1MS FOR INTERRUPT
2019  021352  005737  003012                  TST     DONE            ;TEST IN INTERRUPT FLAG SET
2020  021356  001460                          BEQ     14$             ;NO - SKIP
2021  021360  032737  000001  003050  5$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
2022  021366  001035                          BNE     10$             ;YES - SKIP
2023  021370  012703  010124                  MOV     #MDRDY,R3       ;SET NO READY MESSAGE
2024  021374  012704  011026                  MOV     #CAFDT,R4       ;CONDITION OF AFTER DATA XFER
2025  021400                                  ERRHRD  10017.,,ERR5
 (4)  021400  104456                          TRAP    C$ERHRD
 (5)  021402  023441                          .WORD   10017
 (5)  021404  000000                          .WORD   0
 (5)  021406  012322                          .WORD   ERR5
2026  021410  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
2027  021414  004737  016430          4$:     JSR     PC,GSTAT        ;GET STATUS
2028  021420  021536                          60$
2029  021422  032737  000001  003050          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE HAS COME READY
2030  021430  001403                          BEQ     11$             ;NO - SKIP
2031  021432  005037  003022                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
2032  021436  000411                          BR      10$             ;SKIP
2033  021440  005301                  11$:    DEC     R1              ;DEC WAIT COUNT
2034  021442  001364                          BNE     4$              ;LOOP UNTIL TIME DONE
2035  021444  012704  011037                  MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECONDS
2036  021450                                  ERRHRD  10014.,,ERR5
 (4)  021450  104456                          TRAP    C$ERHRD
 (5)  021452  023436                          .WORD   10014
 (5)  021454  000000                          .WORD   0
 (5)  021456  012322                          .WORD   ERR5
2037  021460  000426                          BR      60$             ;EXIT
2038  021462  005737  003050          10$:    TST     T.CS            ;CHECK FOR ANY ERRORS
2039  021466  100005                          BPL     12$             ;NO - SKIP
2040  021470                                  ERRHRD  10016.,,ERR6    ;REPORT ALL ERRORS
 (4)  021470  104456                          TRAP    C$ERHRD
 (5)  021472  023440                          .WORD   10016
 (5)  021474  000000                          .WORD   0
 (5)  021476  012372                          .WORD   ERR6
2041  021500  000416                          BR      60$
2042  021502  012701  003060          12$:    MOV     #HDWRD2,R1      ;GET POINTER
2043  021506  016221  000006                  MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
2044  021512  016221  000006                  MOV     RLMP(R2),(R1)+
2045  021516  000411                          BR      65$             ;EXIT
2046  021520  004737  016224          14$:    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
2047  021524  012603                          MOV     (SP)+,R3        ;GET RESULTS
2048  021526                                  ERRHRD  10015.,,ERR1    ;REPORT
 (4)  021526  104456                          TRAP    C$ERHRD
 (5)  021530  023437                          .WORD   10015
 (5)  021532  000000                          .WORD   0
 (5)  021534  012070                          .WORD   ERR1
2049  021536  005037  003022          60$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
2050  021542  005737  003132          65$:    TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
2051  021546  001007                          BNE     22$             ;NO - SKIP
2052  021550  012703  003040                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
2053  021554  012701  000004                  MOV     #4,R1           ;SET COUNT
```

```
2054  021560  012623                    20$:    MOV     (SP)+,(R3)+     ;RESTORE REGISTER
2055  021562  005301                            DEC     R1              ;DEC COUNT
2056  021564  001375                            BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
2057  021566  162737  000002  003006    22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2058  021574  012604                            MOV     (SP)+,R4        ;RESTORE REGS
2059  021576  012601                            MOV     (SP)+,R1
2060  021600  012600                            MOV     (SP)+,R0
2061  021602  012603                            MOV     (SP)+,R3
2062  021604  005737  003022                    TST     ERRSWI          ;TEST IF ERROR RETURN
2063  021610  001403                            BEQ     99$             ;YES - SKIP
2064  021612  063716  003022                    ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2065  021616  000207                            RTS     PC
2066  021620  017616  000000            99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2067  021624  000207                            RTS     PC
2068
2070                                      ;       VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
2071                                      ;       SEQUENCE.
2072  021626  010346                    VERHDR: MOV     R3,-(SP)        ;STORE REGS
2073  021630  013703  003006                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2074  021634  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2075  021636  016663  000002  002410            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2076  021644  162763  000004  002410            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2077  021652  010337  003006                    MOV     R3,SSINDX       ;STORE IT BACK
2078  021656  010046                            MOV     R0,-(SP)
2079  021660  010146                            MOV     R1,-(SP)
2080  021662  010446                            MOV     R4,-(SP)
2081  021664  010546                            MOV     R5,-(SP)
2082  021666  012737  000002  003022            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
2083  021674  052737  000002  003010            BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
2084  021702  005037  003020                    CLR     MORECE          ;CLEAR MORE ERRORS FLAG
2085  021706  012704  004072                    MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
2086  021712  012705  003122                    MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
2087  021716  005003                            CLR     R3              ;CLEAR FOR WORD COUNTER
2088  021720  011415                            MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
2089  021722  011401                            MOV     (R4),R1         ;PUT WORD IN REG 1
2090  021724  042701  000177                    BIC     #177,R1 ;CLEAR ALL BUT CYLINDER
2091  021730  012700  000007                    MOV     #7,R0           ;SET SHIFT COUNT
2092  021734  006201                    3$:     ASR     R1              ;SHIFT
2093  021736  005300                            DEC     R0              ;DEC
2094  021740  001375                            BNE     3$              ;LOOP
2095  021742  020137  003106                    CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
2096  021746  001407                            BEQ     4$              ;YES - SKIP
2097  021750                                    ERRHRD  10018.,,ERR10   ;REPORT ERROR
  (4) 021750  104456                            TRAP    C$ERHRD
  (5) 021752  023442                            .WORD   10018
  (5) 021754  000000                            .WORD   0
  (5) 021756  013464                            .WORD   ERR10
2098  021760  005037  003022                    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
2099  021764  000456                            BR      65$
2100  021766  012701  000050            4$:     MOV     #40.,R1         ;SET HEADER COUNT
2101  021772  042715  000100                    BIC     #HDHSEL,(R5)    ;CLEAR HEAD SELECT AND 0 BIT
2102  021776  005737  003116                    TST     DESHD           ;ARE WE USING HD 0?
2103  022002  001402                            BEQ     5$              ;YES - SKIP
2104  022004  052715  000100                    BIS     #HDHSEL,(R5)    ;INSERT HEAD BIT
2105  022010  005065  000002            5$:     CLR     2(R5)           ;CLEAR 2ND WORD OF WORK AREA
2106  022014  021524                    6$:     CMP     (R5),(R4)+      ;TEST FIRST WORD OK
```

```
2107  022016  001410                          BEQ     8$              ;YES - SKIP
2108  022020  005744                          TST     -(R4)           ;ELSE SET POINTER FOR ERROR
2109  022022                                  ERRHRD  10018.,,ERR10   ;REPORT
 (4)  022022  104456                          TRAP    C$ERHRD
 (5)  022024  023442                          .WORD   10018
 (5)  022026  000000                          .WORD   0
 (5)  022030  013464                          .WORD   ERR10
2110  022032  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
2111  022036  005724                          TST     (R4)+           ;RESET POINTER
2112  022040  005203                  8$:     INC     R3              ;BUMP WORD COUNTER
2113  022042  005724                          TST     (R4)+           ;TEST 2ND WORD IS 0
2114  022044  001410                          BEQ     12$             ;YES - SKIP
2115  022046  022544                          CMP     (R5)+,-(R4)     ;ADJUST POINTERS FOR REPORT
2116  022050                                  ERRHRD  10018.,,ERR10   ;REPORT
 (4)  022050  104456                          TRAP    C$ERHRD
 (5)  022052  023442                          .WORD   10018
 (5)  022054  000000                          .WORD   0
 (5)  022056  013464                          .WORD   ERR10
2117  022060  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
2118  022064  024524                          CMP     -(R5),(R4)+     ;RESET POINTERS
2119  022066  005724                  12$:    TST     (R4)+           ;BUMP PAST ECC WORD
2120  022070  005203                          INC     R3              ;BUMP WORD COUNTER
2121  022072  005215                          INC     (R5)            ;BUMP SECTOR OF EXPECTED HEADER
2122  022074  011500                          MOV     (R5),R0         ;MOVE EXPECTED HDR TO R0
2123  022076  042700  177700                  BIC     #^CHDSEC,R0     ;CLEAR ALL BUT SECTOR
2124  022102  022700  000050                  CMP     #40.,R0         ;TEST IF AT SECTOR 40
2125  022106  001002                          BNE     15$             ;NO - SKIP
2126  022110  042715  000077                  BIC     #HDSEC,(R5)     ;CLEAR SECTOR TO 0
2127  022114  005203                  15$:    INC     R3              ;BUMP HDR WORD COUNTER
2128  022116  005301                          DEC     R1              ;DEC HEADER COUNT
2129  022120  001335                          BNE     6$              ;LOOP IF NOT YET DONE
2130  022122  162737  000002  003006  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2131  022130  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
2132  022132  012604                          MOV     (SP)+,R4
2133  022134  012601                          MOV     (SP)+,R1
2134  022136  012600                          MOV     (SP)+,R0
2135  022140  012603                          MOV     (SP)+,R3
2136  022142  005737  003022                  TST     ERRSWI          ;TEST IF ERROR RETURN
2137  022146  001403                          BEQ     99$             ;YES - SKIP
2138  022150  063716  003022                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2139  022154  000207                          RTS     PC
2140  022156  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2141  022162  000207                          RTS     PC
2142
2144                                  ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
2145  022164  013705  003056          POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
2146  022170  000402                          BR      POSHD0          ;SKIP
2147  022172  013705  003056          POSHSB: MOV     T.MP,R5 ;START FOR POSITION HD BIT IN MP
2148  022176  010146          POSHD0: MOV     R1,-(SP)        ;STORE R1
2149  022200  042705  177677                  BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
2150  022204  012701  000006                  MOV     #6,R1           ;SET SHIFT COUNT
2151  022210  006205                  1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
2152  022212  005301                          DEC     R1
2153  022214  001375                          BNE     1$
2154  022216  012601                          MOV     (SP)+,R1        ;RESTORE R1
2155  022220  000207                          RTS     PC              ;RETURN
```

```
                                        ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
                                        ;       FROM THE CALLING ROUTINE IN R1.
                                        ;
022222  010346                          RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
022224  013703  003006                                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
022230  005723                                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
022232  016663  000002  002410                          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
022240  162763  000004  002410                          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
022246  010337  003006                                  MOV     R3,SSINDX       ;STORE IT BACK
022252  010046                                          MOV     R0,-(SP)
022254  010146                                          MOV     R1,-(SP)
022256  010446                                          MOV     R4,-(SP)
022260  012737  000002  003022                          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
022266  004737  016430                  5$:             JSR     PC,GSTAT        ;GET DRIVE STATUS
022272  022442                                          10$
022274  032737  000001  003050                          BIT     #DRDYMSK,T.CS   ;CHECK IF READY
022302  001061                                          BNE     9$              ;YES - SKIP
022304  005301                                          DEC     R1              ;DEC WAIT COUNT
022306  001406                                          BEQ     7$              ;SKIP IF 0
022310                                                  WAITUS  #1
022322  000761                                          BR      5$
022324  012703  010124                  7$:             MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
022330                                                  ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
 (4)    022330  104456                                  TRAP    C$ERHRD
 (5)    022332  023444                                  .WORD   10020
 (5)    022334  000000                                  .WORD   0
 (5)    022336  012204                                  .WORD   ERR3
022340  012701  000062                                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
022344  004737  016430                  6$:             JSR     PC,GSTAT        ;GET DRIVE STATUS
022350  022442                                          10$
022352  032737  000001  003050                          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
022360  001016                                          BNE     8$              ;YES - SKIP
022362                                                  WAITMS  #1              ;WAIT 100 MS
022374  005301                                          DEC     R1              ;DEC WAIT COUNT
022376  001362                                          BNE     6$              ;LOOP UNTIL TIME DONE
022400  012704  011037                                  MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
022404                                                  ERRHRD  10021.,,ERR5
 (4)    022404  104456                                  TRAP    C$ERHRD
 (5)    022406  023445                                  .WORD   10021
 (5)    022410  000000                                  .WORD   0
 (5)    022412  012322                                  .WORD   ERR5
022414  000410                                          BR      11$             ;EXIT
022416  032737  100000  003050  8$:             BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
022424  001406                                          BEQ     10$             ;NO - SKIP
022426                                                  ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
 (4)    022426  104456                                  TRAP    C$ERHRD
 (5)    022430  023446                                  .WORD   10022
 (5)    022432  000000                                  .WORD   0
 (5)    022434  012372                                  .WORD   ERR6
022436  005337  003244          11$:            DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
022442  005037  003022          10$:            CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
022446  162737  000002  003006  9$:             SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
022454  012604                                  MOV     (SP)+,R4        ;RESTORE REGISTERS
022456  012601                                  MOV     (SP)+,R1
022460  012600                                  MOV     (SP)+,R0
022462  012603                                  MOV     (SP)+,R3        ;RESTORE R3
```

```
2200  022464  005737  003022            TST     ERRSWI          ;TEST IF ERROR RETURN
2201  022470  001403                    BEQ     99$             ;YES - SKIP
2202  022472  063716  003022            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2203  022476  000207                    RTS     PC
2204  022500  017616  000000      99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2205  022504  000207                    RTS     PC
2206
2207                                ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
2208                                ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
2209                                ;       NUMBER IN CURCYL.
2210  022506  010346                GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
2211  022510  013703  003006            MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2212  022514  005723                    TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2213  022516  016663  000002  002410    MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2214  022524  162763  000004  002410    SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2215  022532  010337  003006            MOV     R3,SSINDX       ;STORE IT BACK
2216  022536  010046                    MOV     R0,-(SP)
2217  022540  010546                    MOV     R5,-(SP)
2218  022542  004737  021172            JSR     PC,XRDHD        ;DO READ HEADER
2219  022546  022576                    65$
2220  022550  013703  003056            MOV     HDWRD1,R3       ;GET HEADER WORD
2221  022554  012705  000007            MOV     #7,R5           ;SET SHIFT COUNT
2222  022560  006203                4$: ASR     R3              ;SHIFT TO RIGHT JUSTIFY
2223  022562  005305                    DEC     R5
2224  022564  001375                    BNE     4$
2225  022566  042703  177000            BIC     #177000,R3
2226  022572  010337  003110            MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
2227  022576  162737  000002  003006 65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2228  022604  012605                    MOV     (SP)+,R5        ;RESTORE REGISTERS
2229  022606  012600                    MOV     (SP)+,R0
2230  022610  012603                    MOV     (SP)+,R3
2231  022612  005737  003022            TST     ERRSWI          ;TEST IF ERROR RETURN
2232  022616  001403                    BEQ     99$             ;YES - SKIP
2233  022620  063716  003022            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2234  022624  000207                    RTS     PC
2235  022626  017616  000000      99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2236  022632  000207                    RTS     PC
2237
2239                                ;       VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
2240                                ;       CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
2241  022634  010346                VERPOS: MOV     R3,-(SP)        ;STORE R3
2242  022636  013703  003006            MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2243  022642  005723                    TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2244  022644  016663  000002  002410    MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2245  022652  162763  000004  002410    SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2246  022660  010337  003006            MOV     R3,SSINDX       ;STORE IT BACK
2247
2248  022664  012737  000002  003022    MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
2249  022672  004737  022506            JSR     PC,GETPOS       ;GET POSITION
2250  022676  022724                    65$
2251  022700  023737  003106  003110    CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
2252  022706  001406                    BEQ     1$              ;YES - SKIP
2253  022710                            ERRHRD  10022.,,ERR8
 (4)  022710  104456                    TRAP    C$ERHRD
 (5)  022712  023446                    .WORD   10022
 (5)  022714  000000                    .WORD   0
```

```
  (5)  022716  013324                                .WORD   ERR8
2254   022720  005037  003022                        CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
2255   022724                              1$:
2256   022724  162737  000002  003006    65$:      SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2257   022732  012603                              MOV     (SP)+,R3        ;RESTORE R3
2258   022734  005737  003022                      TST     ERRSWI          ;TEST IF ERROR RETURN
2259   022740  001403                              BEQ     99$             ;YES - SKIP
2260   022742  063716  003022                      ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2261   022746  000207                              RTS     PC
2262   022750  017616  000000            99$:      MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2263   022754  000207                              RTS     PC
2264
2266                                      ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
2267                                      ;       IN IBUFF.
2268   022756  010346                      RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
2269   022760  013703  003006                      MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2270   022764  005723                              TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2271   022766  016663  000002  002410              MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2272   022774  162763  000004  002410              SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2273   023002  010337  003006                      MOV     R3,SSINDX       ;STORE IT BACK
2274   023006  010046                              MOV     R0,-(SP)
2275   023010  010146                              MOV     R1,-(SP)
2276   023012  010446                              MOV     R4,-(SP)
2277   023014  012737  000002  003022              MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
2278   023022  012701  000050                      MOV     #40.,R1         ;SET HEADER COUNT
2279   023026  052737  100000  003010              BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
2280   023034  012703  004072                      MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
2281   023040  013704  003032                      MOV     RLBAS,R4        ;GET BASE ADDRESS
2282   023044  062704  000006                      ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
2283   023050  012737  000010  003040              MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
2284   023056  053737  003036  003040              BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
2285   023064  042737  002000  003040              BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2286   023072  005037  003042                      CLR     L.BA            ;CLEAR BA
2287   023076  005037  003044                      CLR     L.DA            ;CLEAR DA
2288   023102  005737  003116                      TST     DESHD           ;TEST IF HEAD 0
2289   023106  001403                              BEQ     3$              ;YES - SKIP
2290   023110  052737  000020  003044              BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
2291   023116  013762  003044  000004    3$:       MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
2292   023124  013762  003042  000002              MOV     L.BA,RLBA(R2)   ;LOAD RLBA
2293   023132  032762  000200  000000              BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
2294   023140  001003                              BNE     6$              ;YES - SKIP
2295   023142  004737  020444                      JSR     PC,RDYCHK       ;ELSE CHECK READY
2296   023146  023264                              65$
2297   023150  013762  003040  000000    6$:       MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
2298   023156  012700  077777                      MOV     #77777,R0       ;SET COUNT FOR WAIT
2299   023162  032762  000200  000000    7$:       BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
2300   023170  001016                              BNE     8$              ;YES - SKIP
2301   023172  005300                              DEC     R0              ;DEC COUNT
2302   023174  001372                              BNE     7$              ;SKIP IF NOT YET 0
2303   023176  004737  016172                      JSR     PC,READRL       ;ELSE GET ALL REGISTERS
2304   023202  004737  016224                      JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
2305   023206  012603                              MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
2306   023210                                      ERRHRD  10025.,,ERR1
  (4)  023210  104456                              TRAP    C$ERHRD
  (5)  023212  023451                              .WORD   10025
  (5)  023214  000000                              .WORD   0
```

```
  (5)  023216  012070                          .WORD   ERR1
2307   023220  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
2308   023224  000417                          BR      65$
2309   023226  005737  003050          8$:     TST     T.CS            ;TEST FOR ANY ERRORS
2310   023232  100007                          BPL     12$             ;NO - SKIP
2311   023234                                  ERRHRD  10026.,,ERR6
  (4)  023234  104456                          TRAP    C$ERHRD
  (5)  023236  023452                          .WORD   10026
  (5)  023240  000000                          .WORD   0
  (5)  023242  012372                          .WORD   ERR6
2312   023244  005037  003022                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
2313   023250  000405                          BR      65$
2314   023252  011423          12$:    MOV     (R4),(R3)+      ;STORE HEADER WORDS
2315   023254  011423                  MOV     (R4),(R3)+
2316   023256  011423                  MOV     (R4),(R3)+
2317   023260  005301                  DEC     R1              ;DEC HEADER COUNT
2318   023262  001332                  BNE     6$
2319   023264  162737  000002  003006  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2320   023272  012604                  MOV     (SP)+,R4        ;RESTORE REGISTERS
2321   023274  012601                  MOV     (SP)+,R1
2322   023276  012600                  MOV     (SP)+,R0
2323   023300  012603                  MOV     (SP)+,R3
2324   023302  005737  003022                  TST     ERRSWI          ;TEST IF ERROR RETURN
2325   023306  001403                          BEQ     99$             ;YES - SKIP
2326   023310  063716  003022                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2327   023314  000207                          RTS     PC
2328   023316  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2329   023322  000207                          RTS     PC
2330
2331
2333                           ;       GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
2334                           ;       IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
2335                           ;       IN OBUFF.
2336   023324  010146          DATGEN: MOV     R1,-(SP)        ;STORE REG.STERS
2337   023326  010346                  MOV     R3,-(SP)
2338   023330  010446                  MOV     R4,-(SP)
2339   023332  012701  004472                  MOV     #OBUFF,R1       ,SET POINTER TO OBUFF
2340   023336  012504                  MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
2341   023340  006304                  ASL     R4              ;ADJUST IT FOR INDEXING
2342   023342  016403  002364                  MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
2343   023346  011321                  MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
2344   023350  001421                  BEQ     5$              ;SKIP IF PATTERN IS 0
2345   023352  021327  177777                  CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
2346   023356  001416                  BEQ     5$              ;YES - SKIP
2347   023360  020427  000010                  CMP     R4,#8.          ;TEST IF PATTERN 5
2348   023364  001403                  BEQ     3$              ;YES - SKIP
2349   023366  020427  000020                  CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
2350   023372  002413                  BLT     6$              ;NO - SKIP
2351   023374  005723          3$:     TST     (R3)+           ;BUMP SOURCE POINTER
2352   023376  012321                  MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
2353   023400  012321                  MOV     (R3)+,(R1)+
2354   023402  012704  000015                  MOV     #13.,R4         ;SET COUNT
2355   023406  012703  004472                  MOV     #OBUFF,R3       ;RESET POINTER
2356   023412  000406                  BR      8$
2357   023414  012703  004472          5$:     MOV     #OBUFF,R3       ;ELSE SET OBUFF AS PATTERN SOURCE
2358   023420  000401                  BR      7$              ;GO TO FILL
```

```
 2359  023422  005723                        6$:     TST     (R3)+           ;BUMP SOURCE POINTER
 2360  023424  012704  000017                7$:     MOV     #15.,R4         ;SET MOVE COUNT
 2361  023430  012321                        8$:     MOV     (R3)+,(R1)+     ;MOVE 15 WORDS INTO BUFFER
 2362  023432  005304                                DEC     R4
 2363  023434  001375                                BNE     8$
 2364  023436  012703  004472                        MOV     #OBUFF,R3       ;SET SOURCE TO TOP OF OBUFF
 2365  023442  012704  000160                        MOV     #112.,R4        ;SET COUNT FOR REST OF BUFFER
 2366  023446  012321                        10$:    MOV     (R3)+,(R1)+     ;REPEAT PATTERN IN BUFFER
 2367  023450  005304                                DEC     R4
 2368  023452  001375                                BNE     10$
 2369  023454  012604                                MOV     (SP)+,R4        ;RESTORE REGISTERS
 2370  023456  012603                                MOV     (SP)+,R3
 2371  023460  012601                                MOV     (SP)+,R1
 2372  023462  000205                                RTS     R5              ;RETURN
 2373
 2374                                  ;       DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
 2375                                  ;       ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
 2376  023464  010346                DATCOM: MOV     R3,-(SP)        ;STORE R3
 2377  023466  013703  003006                MOV     SSINDX,R3       ;GET SUBROUTINE STACK INDEX
 2378  023472  005723                        TST     (R3)+           ;BUMP INDEX TO NEXT ENTRY
 2379  023474  016663  000002  002410        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 2380  023502  162763  000004  002410        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 2381  023510  010337  003006                MOV     R3,SSINDX       ;STORE IT BACK
 2382  023514  010146                        MOV     R1,-(SP)        ;STORE OTHER REGISTERS
 2383  023516  010446                        MOV     R4,-(SP)
 2384  023520  010546                        MOV     R5,-(SP)
 2385  023522  052737  000001  003010        BIS     #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
 2386  023530  005037  003020                CLR     MORECE          ;CLEAR MORE ERROR FLAG
 2387  023534  012705  004472                MOV     #OBUFF,R5       ;SET POINTERS TO DATA FOR COMPARE
 2388  023540  012704  004072                MOV     #IBUFF,R4
 2389  023544  012703  000001                MOV     #1,R3           ;SET WORD COUNTER
 2390  023550  012701  000200                MOV     #128.,R1        ;SET COMPARE COUNT
 2391  023554  022425                5$:     CMP     (R4)+,(R5)+     ;COMPARE DATA
 2392  023556  001052                        BNE     10$             ;ERROR - SKIP TO REPORT
 2393  023560  005203                7$:     INC     R3              ;BUMP WORD COUNT
 2394  023562  005301                        DEC     R1              ;DEC COMPARE COUNT
 2395  023564  001373                        BNE     5$              ;LOOP IF NOT 0
 2396  023566  042737  000001  003010  9$:     BIC     #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
 2397  023574  005737  003022                TST     ERRSWI          ;TEST IF ANY COMPARE ERRORS
 2398  023600  001021                        BNE     15$             ;NO - SKIP
 2399  023602  012701  000200                MOV     #128.,R1        ;SET REPORT VALUE
 2400  023606                                PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
  (11) 023606  010146                        MOV     R1,-(SP)
  (10) 023610  012746  010743                MOV     #RESE6,-(SP)
  (9)  023614  013746  003020                MOV     MORECE,-(SP)
  (8)  023620  012746  007614                MOV     #TCERR,-(SP)
  (7)  023624  012746  012037                MOV     #FMT27,-(SP)
  (6)  023630  012746  000005                MOV     #5,-(SP)
  (3)  023634  010600                        MOV     SP,R0
  (4)  023636  104414                        TRAP    C$PNTB
  (4)  023640  062706  000014                ADD     #14,SP
 2401  023644  162737  000002  003006  15$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 2402  023652  012605                        MOV     (SP)+,R5        ;RESTORE REGS
 2403  023654  012604                        MOV     (SP)+,R4
 2404  023656  012601                        MOV     (SP)+,R1
 2405  023660  012603                        MOV     (SP)+,R3
```

```
2406  023662  005737  003022                TST     ERRSWI          ;TEST IF ERROR RETURN
2407  023666  001403                       BEQ     99$             ;YES - SKIP
2408  023670  063716  003022                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2409  023674  000207                       RTS     PC
2410  023676  017616  000000        99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2411  023702  000207                       RTS     PC
2412  023704  023737  003020  013734  10$:  CMP     MORECE,DCLIMW   ;TEST IF COMPARE EPRORS LIMIT EXCEEDED
2413  023712  002011                       BGE     13$             ;YES - SKIP
2414  023714  024445                       CMP     -(R4),-(R5)     ;SET PTRS BACK TO ERROR WORDS
2415  023716                               ERRHRD  10035.,,ERR10   ;REPORT ERROR
 (4)  023716  104456                       TRAP    C$ERHRD
 (5)  023720  023463                       .WORD   10035
 (5)  023722  000000                       .WORD   0
 (5)  023724  013464                       .WORD   ERR10
2416  023726  005037  003022                CLR     ERRSWI          ;CLEAR ERROR SWITCH
2417  023732  022425                       CMP     (R4)+,(R5)+     ;BUMP PTRS PAST ERROR WORDS
2418  023734  000711                       BR      7$              ;DO NEXT COMPARE
2419  023736  005237  003020        13$:    INC     MORECE          ;BUMP ERROR COUNTER
2420  023742  000706                       BR      7$              ;DO NEXT COMPARE
```

```
2422
2423                                        ;       WRITE AND READ DATA ROUTINE.
2424
2425  023744  012737  177777  003124  XWRITT: MOV     #-1,TEMP1       ;SET SPECIAL WRITE FOR TIMING FLAG
2426  023752  000402                          BR      XWRIT1
2427  023754  005037  003124          XWRITE: CLR     TEMP1           ;CLEAR SPECIAL WRITE FLAG
2428  023760  012737  000112  003140  XWRIT1: MOV     #WTDATA,TEMP7   ;SET FOR WRITE
2429  023766  023737  002306  003110          CMP     HLMTW,CURCYL    ;TEST IF CYLINDER 255 (BAD SEC)
2430  023774  001006                          BNE     1$              ;NO - SKIP
2431  023776  005737  003116                  TST     DESHD           ;TEST IF HEAD 1 (BAD SECTOR FILES)
2432  024002  001403                          BEQ     1$              ;NO - SKIP
2433  024004  052737  004000  003010          BIS     #BADADD,OPFLAG  ;SET BAD ADDRESS FLAG
2434  024012  000403                  1$:     BR      XREADG          ;SKIP TO EXECUTE
2435  024014  012737  000114  003140  XREAD:  MOV     #RDDATA,TEMP7   ;SET FOR READ
2436  024022  010346                  XREADG: MOV     R3,-(SP)        ;STORE R3
2437  024024  013703  003006                  MOV     SSINDX,R3       ;SET SUBROUTINE INDEX
2438  024030  005723                          TST     (R3)+           ;BUMP TO NEXT STACK ENTRY
2439  024032  016663  000002  002410          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2440  024040  162763  000004  002410          SUB     #4,SUBSTK(R3)   ;ADJUST TO POINT TO CALL
2441  024046  010337  003006                  MOV     R3,SSINDX       ;STORE IT BACK
2442  024052  010046                          MOV     R0,-(SP)
2443  024054  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
2444  024056  010446                          MOV     R4,-(SP)
2445  024060  004737  020444                  JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
2446  024064  024452                          65$
2447  024066  012703  003040                  MOV     #L.CS,R3        ;GET ADDRESS OF LOAD REGS
2448  024072  013713  003140                  MOV     TEMP7,(R3)      ;SET COMMAND
2449  024076  053713  003036                  BIS     RLDRV,(R3)      ;INSERT DRIVE NUMBER
2450  024102  042713  002000                  BIC     #BIT10,(R3)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2451  024106  032723  000004                  BIT     #BIT2,(R3)+     ;TEST IF WRITE DATA
2452  024112  001403                          BEQ     3$              ;YES - SKIP
2453  024114  012723  004072                  MOV     #IBUFF,(R3)+    ;ELSE SET BA FOR READ
2454  024120  000402                          BR      4$
2455  024122  012723  004472          3$:     MOV     #OBUFF,(R3)+    ;SET BA FOR WRITE
2456  024126  013713  003110          4$:     MOV     CURCYL,(R3)     ;GET CURRENT CYLINDER
2457  024132  012704  000007                  MOV     #7,R4           ;ALIGN IT IN DA
2458  024136  006313                  5$:     ASL     (R3)
2459  024140  005304                          DEC     R4
2460  024142  001375                          BNE     5$
2461  024144  005737  003116                  TST     DESHD           ;TEST IF HEAD 0
2462  024150  001402                          BEQ     7$              ;YES - SKIP
2463  024152  052713  000100                  BIS     #HSMSK,(R3)     ;SET FOR HEAD 1
2464  024156  053723  003120          7$:     BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
2465  024162  012713  177600                  MOV     #177600,(R3)    ;INSERT WORD COUNT
2466  024166  005737  003124                  TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
2467  024172  001402                          BEQ     8$              ;NO - SKIP
2468  024174  012713  177777                  MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
2469  024200  032737  004000  003010  8$:     BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
2470  024206  001414                          BEQ     2$              ;NO - SKIP
2471  024210  042737  173777  003010          BIC     #^CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
2472  024216  012703  010645                  MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
2473  024222                                  ERRHRD  10032.,,ERR1
 (4)  024222  104456                          TRAP    C$ERHRD
 (5)  024224  023460                          .WORD   10032
 (5)  024226  000000                          .WORD   0
 (5)  024230  012070                          .WORD   ERR1
```

```
 2474  024232  005037  003010                 CLR     OPFLAG          ;CLEAR ALL FLAGS
 2475  024236  000503                         BR      64$
 2476  024240  005037  003012         2$:     CLR     DONE            ;CLEAR INTERRUPT FLAG
 2477  024244  005737  003124                 TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
 2478  024250  001100                         BNE     65$             ;YES - DO NOT START WRITE
 2479  024252  011362  000006                 MOV     (R3),RLMP(R2)   ;LOAD RL REGS
 2480  024256  014362  000004                 MOV     -(R3),RLDA(R2)
 2481  024262  014362  000002                 MOV     -(R3),RLBA(R2)
 2482  024266  014362  000000                 MOV     -(R3),RLCS(R2)
 2483  024272                         10$:    WAITUS  #3000.          ;WAIT 300MS FOR INTERRUPT
 2484  024304  005737  003012                 TST     DONE            ;CHECK IF INTERRUPT
 2485  024310  001010                         BNE     14$             ;YES - SKIP
 2486  024312  004737  016224                 JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 2487  024316  012603                         MOV     (SP)+,R3        ;GET RESULT MESSAGE
 2488  024320                                 ERRHRD  10030.,,ERR1
   (4) 024320  104456                         TRAP    C$ERHRD
   (5) 024322  023456                         .WORD   10030
   (5) 024324  000000                         .WORD   0
   (5) 024326  012070                         .WORD   ERR1
 2489  024330  000446                         BR      64$
 2490  024332  032737  000001  003050 14$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 2491  024340  001033                         BNE     20$             ;YES - SKIP
 2492  024342  012703  010124                 MOV     #MDRDY,R3       ;SET RESULT MESSAGE
 2493  024346  012704  011026                 MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
 2494  024352                                 ERRHRD  10032.,,ERR5
   (4) 024352  104456                         TRAP    C$ERHRD
   (5) 024354  023460                         .WORD   10032
   (5) 024356  000000                         .WORD   0
   (5) 024360  012322                         .WORD   ERR5
 2495  024362  012701  000062                 MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
 2496  024366  004737  016430         17$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
 2497  024372  024446                         64$
 2498  024374  032737  000001  003050         BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
 2499  024402  001012                         BNE     20$             ;YES - SKIP
 2500  024404  005301                         DEC     R1              ;DEC WAIT COUNT
 2501  024406  001367                         BNE     17$             ;LOOP IF NOT TIME DONE
 2502  024410  012704  011037                 MOV     #C5SEC,R4       ;SET CONDITION 5 SECONDS
 2503  024414                                 ERRHRD  10033.,,ERR5
   (4) 024414  104456                         TRAP    C$ERHRD
   (5) 024416  023461                         .WORD   10033
   (5) 024420  000000                         .WORD   0
   (5) 024422  012322                         .WORD   ERR5
 2504  024424  005037  003022                 CLR     ERRSWI          ;CLEAR ERROR SWITCH
 2505  024430  005737  003050         20$:    TST     T.CS            ;CHECK IF ANY ERROR
 2506  024434  100006                         BPL     65$             ;NO - SKIP
 2507  024436                                 ERRHRD  10031.,,ERR6
   (4) 024436  104456                         TRAP    C$ERHRD
   (5) 024440  023457                         .WORD   10031
   (5) 024442  000000                         .WORD   0
   (5) 024444  012372                         .WORD   ERR6
 2508  024446  005037  003022         64$:    CLR     ERRSWI          ;CLEAR ERROR SWITCH
 2509  024452  162737  000002  003006 65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 2510  024460  012604                         MOV     (SP)+,R4        ;RESTORE REGISTERS
 2511  024462  012601                         MOV     (SP)+,R1
 2512  024464  012600                         MOV     (SP)+,R0
 2513  024466  012603                         MOV     (SP)+,R3
```

```
2514  024470  005737  003022            TST     ERRSWI          ;TEST IF ERROR RETURN
2515  024474  001403                    BEQ     99$             ;YES - SKIP
2516  024476  063716  003022            ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
2517  024502  000207                    RTS     PC
2518  024504  017616  000000    99$:    MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
2519  024510  000207                    RTS     PC
2520
2521                                    BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
2522                                    DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
2523  024512  010046            BSCHK:  MOV     R0,-(SP)        ;STORE REGISTERS
2524  024514  010146                    MOV     R1,-(SP)
2525  024516  010346                    MOV     R3,-(SP)
2526  024520  005037  003024            CLR     BSFLAG          ;CLEAR FLAG
2527  024524  012703  003676            MOV     #FBSFIL,R3      ;GET POINTER TO FACTORY FILE
2528  024530  022713  177777            CMP     #-1,(R3)        ;CHECK IF ALL ONES
2529  024534  001005                    BNE     4$              ;NO SKIP TO TEST
2530  024536  012703  003502    2$:     MOV     #SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILF
2531  024542  022713  177777            CMP     #-1,(R3)        ;CHECK IF ALL ONES
2532  024546  001431                    BEQ     20$             ;YES - EXIT
2533  024550  013700  003106    4$:     MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
2534  024554  012701  000007            MOV     #7,R1           ;POSITION CYLINDER
2535  024560  006300            5$:     ASL     R0
2536  024562  005301                    DEC     R1
2537  024564  001375                    BNE     5$
2538  024566  005737  003116            TST     DESHD           ;CHECK IF HEAD 0
2539  024572  001402                    BEQ     7$              ;YES - SKIP
2540  024574  052700  000100            BIS     #BIT6,R0        ;INSERT HEAD 1
2541  024600  053700  003120    7$:     BIS     DESSEC,R0       ;INSERT SECTOR
2542  024604  022300    8$:     CMP     (R3)+,R0        ;CHECK THIS WORD IN FILE
2543  024606  001402                    BEQ     12$             ;YES - EXIT,ERROR
2544  024610  101005                    BHI     15$             ;EXIT- NO ERROR
2545  024612  000774                    BR      8$
2546  024614  012737  000001  003024  12$:     MOV     #1,BSFLAG       ;SET ERROR FLAG
2547  024622  000403                    BR      20$             ;GO TO EXIT
2548  024624  020327  003676    15$:    CMP     R3,#FBSFIL      ;DONE BOTH FILES?
2549  024630  003342                    BGT     2$              ;NO GO DO SOFTWARE FILE
2550  024632  012603            20$:    MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
2551  024634  012601                    MOV     (SP)+,R1
2552  024636  012600                    MOV     (SP)+,R0
2553  024640  005737  0C3024            TST     BSFLAG          ;CHECK IF ERROR
2554  024644  001003                    BNE     99$             ;YES - SKIP
2555  024646  062716  000002            ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
2556  024652  000207                    RTS     PC
2557  024654  017616  000000    99$:    MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
2558  024660  000207                    RTS     PC
2559
2561                            ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
2562                            ;       OPERATION BEING PERFORMED PORTION OF ALL
2563                            ;       ERROR MESSAGES.
2564  024662  010446            RPTOP:  MOV     R4,-(SP)
2565  024664  005737  003006            TST     SSINDX          ;TEST SUBROUTINE INDEX 0
2566  024670  001433                    BEQ     1$              ;SKIP IF 0
2567  024672  012704  000002            MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
2568  024676                            PRINTB  #FMT9,#SEQMES   ;PRINT 'SUBROUTINE CALL SEQ''
 (8)  024676  012746  007504            MOV     #SEQMES,-(SP)
 (7)  024702  012746  011356            MOV     #FMT9,-(SP)
```

```
   (6)  024706  012746  000002                    MOV     #2,-(SP)
   (3)  024712  010600                            MOV     SP,R0
   (4)  024714  104414                            TRAP    C$PNTB
   (4)  024716  062706  000006                    ADD     #6,SP
2569    024722                          3$:       PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
   (8)  024722  016446  002410                    MOV     SUBSTK(R4),-(SP)
   (7)  024726  012746  011531                    MOV     #FMT16,-(SP)
   (6)  024732  012746  000002                    MOV     #2,-(SP)
   (3)  024736  010600                            MOV     SP,R0
   (4)  024740  104414                            TRAP    C$PNTB
   (4)  024742  062706  000006                    ADD     #6,SP
2570    024746  062704  000002                    ADD     #2,R4           ;BUMP INDEX
2571    024752  020437  003006                    CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
2572    024756  003761                            BLE     3$              ;LOOP IF NOT ALL PRINTED YET
2573    024760                          1$:       PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
   (9)  024760  012746  006471                    MOV     #TSTLAB,-(SP)
   (8)  024764  013746  003016                    MOV     ERHEAD,-(SP)
   (7)  024770  012746  011161                    MOV     #FMT4,-(SP)
   (6)  024774  012746  000003                    MOV     #3,-(SP)
   (3)  025000  010600                            MOV     SP,R0
   (4)  025002  104414                            TRAP    C$PNTB
   (4)  025004  062706  000010                    ADD     #10,SP
2574    025010  042737  030000  003010            BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
2575    025016  013701  003040                    MOV     L.CS,R1         ;GET COMMAND EXECUTED
2576    025022  042701  177741                    BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
2577    025026  022701  000006                    CMP     #6,R1   ;TEST IF SEEK OPERATION
2578    025032  001003                            BNE     2$              ;NO - SKIP
2579    025034  052737  010000  003010            BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
2580    025042  022701  000012            2$:     CMP     #12,R1          ;TEST IF WRITE
2581    025046  001003                            BNE     20$             ;NO - SKIP
2582    025050  052737  020000  003010            BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
2583    025056  022701  000014            20$:    CMP     #14,R1          ;TEST IF READ
2584    025062  001003                            BNE     22$             ;NO - SKIP
2585    025064  052737  020000  003010            BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
2586    025072                          22$:      PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
   (9)  025072  016146  002230                    MOV     OPMSGS(R1),-(SP)
   (8)  025076  012746  005517                    MOV     #MOPER,-(SP)
   (7)  025102  012746  011137                    MOV     #FMT1,-(SP)
   (6)  025106  012746  000003                    MOV     #3,-(SP)
   (3)  025112  010600                            MOV     SP,R0
   (4)  025114  104414                            TRAP    C$PNTB
   (4)  025116  062706  000010                    ADD     #10,SP
2587    025122  020127  000004                    CMP     R1,#4           ;CHECK IF GET STATUS
2588    025126  001007                            BNE     4$              ;NO - SKIP
2589    025130  032737  000010  003044            BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
2590    025136  001403                            BEQ     4$              ;NO - SKIP
2591    025140  012701  000016                    MOV     #16,R1          ;SET TO PRINT WITH RESET
2592    025144  000436                            BR      9$
2593    025146  032737  007777  003010    4$:     BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
2594    025154  001424                            BEQ     8$              ;NO - SKIP
2595    025156  013704  003010                    MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
2596    025162  012701  000020                    MOV     #20,R1          ;PRESET THE POINTER
2597    025166  032704  000001            5$:     BIT     #BIT00,R4       ;CHECK THE BIT
2598    025172  001003                            BNE     6$              ;IF SET - SKIP
2599    025174  005721                            TST     (R1)+           ;BUMP POINTER
2600    025176  006204                            ASR     R4
```

```
2601  025200  000772                          BR      5$
2602  025202                          6$:     PRINTB  #FMT2,OPMSGS(R1)
 (8)  025202  016146  002230                  MOV     OPMSGS(R1),-(SP)
 (7)  025206  012746  011153                  MOV     #FMT2,-(SP)
 (6)  025212  012746  000002                  MOV     #2,-(SP)
 (3)  025216  010600                          MOV     SP,R0
 (4)  025220  104414                          TRAP    C$PNTB
 (4)  025222  062706  000006                  ADD     #6,SP
2603  025226  032737  100000  003010  8$:     BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
2604  025234  001415                          BEQ     10$             ;NO - SKIP
2605  025236  012701  000050                  MOV     #50,R1          ;ELSE PRINT IT
2606  025242                          9$:     PRINTB  #FMT2,OPMSGS(R1)
 (8)  025242  016146  002230                  MOV     OPMSGS(R1),-(SP)
 (7)  025246  012746  011153                  MOV     #FMT2,-(SP)
 (6)  025252  012746  000002                  MOV     #2,-(SP)
 (3)  025256  010600                          MOV     SP,R0
 (4)  025260  104414                          TRAP    C$PNTB
 (4)  025262  062706  000006                  ADD     #6,SP
2607  025266  000434                          BR      15$             ;SKIP
2608  025270  032737  010000  003010  10$:    BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
2609  025276  001430                          BEQ     15$             ;NO - SKIP
2610  025300                                  PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
 (15) 025300  013746  003116                  MOV     DESHD,-(SP)
 (14) 025304  012746  007445                  MOV     #HDWD,-(SP)
 (13) 025310  013746  003114                  MOV     DESSGN,-(SP)
 (12) 025314  012746  007440                  MOV     #SGNWD,-(SP)
 (11) 025320  013746  003112                  MOV     DFSDIF,-(SP)
 (10) 025324  012746  007432                  MOV     #DIFWD,-(SP)
 (9)  025330  013746  003104                  MOV     OLDCYL,-(SP)
 (8)  025334  012746  007463                  MOV     #FRMWD,-(SP)
 (7)  025340  012746  011377                  MOV     #FMT13,-(SP)
 (6)  025344  012746  000011                  MOV     #11,-(SP)
 (3)  025350  010600                          MOV     SP,R0
 (4)  025352  104414                          TRAP    C$PNTB
 (4)  025354  062706  000024                  ADD     #24,SP
2611  025360  032737  020000  003010  15$:    BIT     #RORWOP,OPFLAG  ;TEST IF READ OR WRITE SET
2612  025366  001424                          BEQ     17$             ;NO - SKIP
2613  025370                                  PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
 (13) 025370  013746  003120                  MOV     DESSEC,-(SP)
 (12) 025374  012746  007451                  MOV     #SECWD,-(SP)
 (11) 025400  013746  003116                  MOV     DESHD,-(SP)
 (10) 025404  012746  007445                  MOV     #HDWD,-(SP)
 (9)  025410  013746  003110                  MOV     CURCYL,-(SP)
 (8)  025414  012746  007456                  MOV     #CYLWD,-(SP)
 (7)  025420  012746  011726                  MOV     #FMT22,-(SP)
 (6)  025424  012746  000007                  MOV     #7,-(SP)
 (3)  025430  010600                          MOV     SP,R0
 (4)  025432  104414                          TRAP    C$PNTB
 (4)  025434  062706  000020                  ADD     #20,SP
2614  025440  004737  026112          17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
2615  025444  012604                          MOV     (SP)+,R4        ;RESTORE R4
2616  025446  000207                          RTS     PC
2617
2618                                  ;       REPORT REASON ROUTINE
2619                                  ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
2620  025450  010146                  RPTRES: MOV     R1,-(SP)        ;STORE R1
```

```
2621  025452  010346                        MOV     R3,-(SP)          ;STORE R3
2622  025454  010446                        MOV     R4,-(SP)          ;STORE R4
2623  025456  012701  003066                MOV     #RESPARM,R1       ;GET START OF PARAM
2624  025462  012103                        MOV     (R1)+,R3          ;GET NUMBER OF PARAM
2625  025464                                PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
 (9)  025464  011146                        MOV     (R1),-(SP)
 (8)  025466  012746  005526                MOV     #MRSLT,-(SP)
 (7)  025472  012746  011144                MOV     #FMT1.1,-(SP)
 (6)  025476  012746  000003                MOV     #3,-(SP)
 (3)  025502  010600                        MOV     SP,R0
 (4)  025504  104414                        TRAP    C$PNTB
 (4)  025506  062706  000010                ADD     #10,SP
2626  025512  021127  010516                CMP     (R1),#MNDRST      ;TEST IF MESSAGE IS NO DRV STATUS
2627  025516  001453                        BEQ     6$                ;YES - SKIP REST OF REPORT
2628  025520  012704  011363                MOV     #FMT11,R4         ;PRISET FOR FORMAT 11
2629  025524  022127  010511                CMP     (R1)+,#MCYLOC     ;CHECK IF REPORTING CYLINDER LOC
2630  025530  001002                        BNE     3$                ;NO - SKIP
2631  025532  012704  011371                MOV     #FMT12,R4         ;ELSE CHANGE TO FORMAT 12
2632  025536  005303              3$:       DEC     R3                ;DEC PARAM COUNT
2633  025540  001442                        BEQ     6$                ;IF 0 - EXIT
2634  025542                                PRINTB  R4,#RESE3,(R1)+   ;REPORT IS VALUE
 (9)  025542  012146                        MOV     (R1)+,-(SP)
 (8)  025544  012746  010725                MOV     #RESE3,-(SP)
 (7)  025550  010446                        MOV     R4,-(SP)
 (6)  025552  012746  000003                MOV     #3,-(SP)
 (3)  025556  010600                        MOV     SP,R0
 (4)  025560  104414                        TRAP    C$PNTB
 (4)  025562  062706  000010                ADD     #10,SP
2635  025566                                PRINTB  R4,#RESE4,(R1)+   ;REPORT SB VALUE
 (9)  025566  012146                        MOV     (R1)+,-(SP)
 (8)  025570  012746  010731                MOV     #RESE4,-(SP)
 (7)  025574  010446                        MOV     R4,-(SP)
 (6)  025576  012746  000003                MOV     #3,-(SP)
 (3)  025602  010600                        MOV     SP,R0
 (4)  025604  104414                        TRAP    C$PNTB
 (4)  025606  062706  000010                ADD     #10,SP
2636  025612  162703  000002                SUB     #2,R3             ;DEC PARAM COUNT
2637  025616  001413                        BEQ     6$                ;IF 0 - EXIT
2638  025620                                PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
 (9)  025620  012146                        MOV     (R1)+,-(SP)
 (8)  025622  012746  010736                MOV     #RESE5,-(SP)
 (7)  025626  012746  011137                MOV     #FMT1,-(SP)
 (6)  025632  012746  000003                MOV     #3,-(SP)
 (3)  025636  010600                        MOV     SP,R0
 (4)  025640  104414                        TRAP    C$PNTB
 (4)  025642  062706  000010                ADD     #10,SP
2639  025646  012604              6$:       MOV     (SP)+,R4          ;RESTORE REGS
2640  025650  012603                        MOV     (SP)+,R3
2641  025652  012601                        MOV     (SP)+,R1
2642  025654  000207                        RTS     PC                ;RETURN
2643
2644                                ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
2645                                ;       AND ALL REGISTER CONTENTS.
2646  025656                        RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 025656  005046                        CLR     -(SP)
 (11) 025660  153716  003037                BISB    RLDRV+1,(SP)
```

```
  (10) 025664  012746  006142                    MOV     #DRVNAM,-(SP)
  (9)  025670  013746  003032                    MOV     RLBAS,-(SP)
  (8)  025674  012746  006131                    MOV     #BASADD,-(SP)
  (7)  025700  012746  011172                    MOV     #FMT5,-(SP)
  (6)  025704  012746  000005                    MOV     #5,-(SP)
  (3)  025710  010600                            MOV     SP,R0
  (4)  025712  104414                            TRAP    C$PNTB
  (4)  025714  062706  000014                    ADD     #14,SP
2647                                    ;        REPORT RL11 REGISTERS
2648  025720                                     PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
  (13) 025720  012746  007445                    MOV     #HDWD,-(SP)
  (12) 025724  012746  007456                    MOV     #CYLWD,-(SP)
  (11) 025730  012746  006245                    MOV     #MPNAM,-(SP)
  (10) 025734  012746  006233                    MOV     #BANAM,-(SP)
  (9)  025740  012746  006240                    MOV     #DANAM,-(SP)
  (8)  025744  012746  006226                    MOV     #CSNAM,-(SP)
  (7)  025750  012746  011212                    MOV     #FMT6,-(SP)
  (6)  025754  012746  000007                    MOV     #7,-(SP)
  (3)  025760  010600                            MOV     SP,R0
  (4)  025762  104414                            TRAP    C$PNTB
  (4)  025764  062706  000020                    ADD     #20,SP
2649  025770                                     PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
  (12) 025770  013746  003046                    MOV     L.MP,-(SP)
  (11) 025774  013746  003042                    MOV     L.BA,-(SP)
  (10) 026000  013746  003044                    MOV     L.DA,-(SP)
  (9)  026004  013746  003040                    MOV     L.CS,-(SP)
  (8)  026010  012746  006252                    MOV     #LAB1,-(SP)
  (7)  026014  012746  011324                    MOV     #FMT8,-(SP)
  (6)  026020  012746  000006                    MOV     #6,-(SP)
  (3)  026024  010600                            MOV     SP,R0
  (4)  026026  104414                            TRAP    C$PNTB
  (4)  026030  062706  000016                    ADD     #16,SP
2650  026034                                     PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
  (14) 026034  013746  003116                    MOV     DESHD,-(SP)
  (13) 026040  013746  003110                    MOV     CURCYL,-(SP)
  (12) 026044  013746  003056                    MOV     T.MP,-(SP)
  (11) 026050  013746  003052                    MOV     T.BA,-(SP)
  (10) 026054  013746  003054                    MOV     T.DA,-(SP)
  (9)  026060  013746  003050                    MOV     T.CS,-(SP)
  (8)  026064  012746  006265                    MOV     #LAB2,-(SP)
  (7)  026070  012746  011254                    MOV     #FMT7,-(SP)
  (6)  026074  012746  000010                    MOV     #10,-(SP)
  (3)  026100  010600                            MOV     SP,R0
  (4)  026102  104414                            TRAP    C$PNTB
  (4)  026104  062706  000022                    ADD     #22,SP
2651  026110  000207                            RTS     PC
2652
2653                                    ;        CLEAR PARAMETER BLOCK FOR REPORTING
2654  026112  010546                   CLRPARM: MOV     R5,-(SP)        ;STORE R5
2655  026114  012701  003066                    MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
2656  026120  012705  000005                    MOV     #5,R5           ;SET COUNT
2657  026124  005021                   2$:      CLR     (R1)+           ;CLEAR WORD
2658  026126  005305                            DEC     R5              ;DEC COUNT
2659  026130  001375                            BNE     2$              ;LOOP UNTIL 0
2660  026132  012701  003066                    MOV     #RESPARM,R1     ;RESET POINTER
2661  026136  012605                            MOV     (SP)+,R5        ;RESTORE R5
```

```
2662  026140  000207                          RTS     PC
2663
2664  026142                          ENDMOD
2665
```

```
                                        .TITLE  CZRLNA0 RL01/02 DRIVE TEST 3
                                        BGNMOD  HRDWTST
                                        .SBTTL  *TEST 1         **SEEK TIMING
                                        BGNTST                  ;TEST 1
                                                                                T1::
                        MOV     #P2T12E,ERHEAD  ;SET ERROR HEADER
                ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
                        TST     CLKFLG          ;P-CLOCK?
                        BNE     3$              ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
                        PRINTF  #FMT9,#NOTST1   ;ELSE, PRINT MSG. 'TEST 1 CANNOT BE PERFORMED...
                        MOV     #NOTST1,-(SP)
                        MOV     #FMT9,-(SP)
                        MOV     #2,-(SP)
                        MOV     SP,R0
                        TRAP    C$PNTF
                        ADD     #6,SP
                                                ;/P-CLOCK IS NOT AVAILABLE''
                        JMP     65$             ;EXIT TEST
                3$:     JSR     PC,TSTINT       ;INITIALIZE TEST
                        JSR     PC,GSTATR       ;CLEAR DRIVE
                        65$
                        MOV     #OFIN,R0        ;GET ADDRESS OF 1ST TIME VALUE
                        MOV     #24.,R1         ;SET COUNT FOR CLEAR
                4$:     CLR     (R0)+           ;CLEAR TIMER STORAGE
                        DEC     R1
                        BNE     4$
                        CLR     PASCNT          ;CLEAR PASS COUNTER
                        CLR     NEWCYL          ;POSITION HEADS AT 0
                        JSR     PC,XSEEK        ;DO SEEK
                        65$
                        MOV     #3000.,R1       ;SET WAIT FOR 300 MS
                        JSR     PC,RDYWAIT      ;WAIT FOR READY
                        65$
                        JSR     PC,VERPOS       ;VERIFY POSITION
                        65$
                        JSR     PC,CHOSHD       ;GO CHOSE HEAD
                        MOV     #OFOUT,R0       ;SET PTRS FOR 1 CYL FWD OUTER TIMER
                        MOV     #OFOUTU,R1
                        MOV     #OROUT,R3
                        MOV     #OROUTU,R4
                        MOV     #1,NEWCYL       ;SET NEWCYL TO CYL 1
                8$:     MOV     #128.,COUNT     ;SET COUNTER FOR SEEK LOOP
                        MOV     #RDHEAD,TEMP8   ;BUILD READ HEADER COMMAND
                        BIS     RLDRV,TEMP8
                        BIC     #BIT10,TEMP8
                9$:     JSR     PC,XSEEKT       ;DO SEEK BUILD BUT DO NOT START
                        65$
                        MOV     L.DA,RLDA(R2)   ;LOAD RL REGISTERS
                        MOV     L.CS,RLCS(R2)
                        MOV     R0,-(SP)        ;STORE R0
                        WAITUS  #10.            ;WAIT FOR INTERRUPT
                        TST     DONE            ;TEST IF INTERRUPT
                        BNE     17$             ;YES - SKIP
```

```
2716  026422  004737  016224            JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
2717  026426  012603                    MOV     (SP)+,R3        ;GET MESSAGE POINTER
2718  026430                            ERRHRD  1201.,,ERR1
 (4)  026430  104456                    TRAP    C$ERHRD
 (5)  026432  002261                    .WORD   1201
 (5)  026434  000000                    .WORD   0
 (5)  026436  012070                    .WORD   ERR1
2719  026440  000137  030052            JMP     65$
2720  026444  005737  003050    17$:    TST     T.CS            ;CHECK IF ANY ERRORS
2721  026450  100006                    BPL     14$             ;NO - SKIP
2722  026452                            ERRHRD  1202 ,,ERR6
 (4)  026452  104456                    TRAP    C$ERHRD
 (5)  026454  002262                    .WORD   1202
 (5)  026456  000000                    .WORD   0
 (5)  026460  012372                    .WORD   ERR6
2723  026462  000137  030052            JMP     65$
2724  026466  005037  003012    14$:    CLR     DONE            ;CLEAR INTERRUPT FLAG
2725  026472                            STCLK                   ;START P-CLOCK TO INITIATE MEASUREMENT
2726                                                            ;/OF TIME INTERVAL
2727  026510  013762  003142  000000    MOV     TEMP8,RLCS(R2)  ;LOAD RL11 CONTROL AND STATUS REGISTER
2728                                                            ;/TO INITIATE SEEK OPERATION
2729  026516                            WAITUS  #2000.          ;WAIT FOR INTERRUPT
2730  026530                            GETTIM  R5              ;GET ELAPSED TIME
2731  026540  012600                    MOV     (SP)+,R0        ;RESTORE R0
2732  026542  013737  003142  003040    MOV     TEMP8,L.CS      ;SET IF ERROR TO REPORT
2733  026550  004737  022634            JSR     PC,VERPOS       ;VERIFY POSITION
2734  026554  030052                    65$
2735  026556  005737  003114            TST     DESSGN          ;CHECK WHICH SEEK DIRECTION
2736  026562  001403                    BEQ     15$             ;REVERSE - SKIP
2737  026564  060510                    ADD     R5,(R0)         ;ADD TO FORWARD TOTAL
2738  026566  005511                    ADC     (R1)            ;ADD IN OVERFLOW
2739  026570  000402                    BR      16$             ;SKIP
2740  026572  060513            15$:    ADD     R5,(R3)         ;ADD TO REVERSE TOTAL
2741  026574  005514                    ADC     (R4)            :ADD IN OVERFLOW
2742  026576  005337  0C3240    16$:    DEC     COUNT           ;DEC SEEK COUNT
2743  026602  001403                    BEQ     18$             ;SKIP IF 0
2744  026604  004737  021004            JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
2745  026610  000662                    BR      9$              ;REDO SEEK LOOP
2746  026612  162710  000470    18$:    SUB     #312.,(R0)      ;SUB CONSTANT FOR READ HEADER TIME
2747  026616  162713  000470            SUB     #312.,(R3)
2748  026622  012705  000006            MOV     #6,R5           ;SET SHIFT COUNT TO DIVIDE BY 64
2749  026626  000241            10$:    CLC                     ;DIVIDE BOTH TOTALS BY 64
2750  026630  006011                    ROR     (R1)
2751  026632  006010                    ROR     (R0)
2752  026634  000241                    CLC
2753  026636  006014                    ROR     (R4)
2754  026640  006013                    ROR     (R3)
2755  026642  005305                    DEC     R5
2756  026644  001370                    BNE     10$
2757  026646  005237  003236            INC     PASCNT          ;BUMP PASS COUNT
2758  026652  022737  000001  003236    CMP     #1,PASCNT       ;TEST IF PASS 1
2759  026660  001051                    BNE     24$             ;NO - SKIP
2760  026662  012737  000177  003106    MOV     #127.,NEWCYL    ;ELSE SET TO POSITION HDS TO 127
2761  026670  022737  000001  002302    CMP     #1,T.DRIVE      ;DRIVE = RL01?
2762  026676  001403                    BEQ     101$            ;YUP
2763  026700  012737  000377  003106    MOV     #255.,NEWCYL    ;NO - SET FOR A MID POS SEEK RL02
```

```
2764  026706  004737  017326          101$:  JSR     PC,XSEEK        ;DO SEEK
2765  026712  030052                         65$
2766  026714  012701  005670                 MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
2767  026720  004737  022222                 JSR     PC,RDYWAIT      ;WAIT FOR READY
2768  026724  030052                         65$
2769  026726  004737  022634                 JSR     PC,VERPOS       ;VERIFY POSITION
2770  026732  030052                         65$
2771  026734  012700  003150                 MOV     #OFMID,R0       ;SET PTRS FOR TIMING 1 CYL SK AT 127
2772  026740  012701  003152                 MOV     #OFMIDU,R1
2773  026744  012703  003164                 MOV     #ORMID,R3
2774  026750  012704  003166                 MOV     #ORMIDU,R4
2775  026754  012737  000200  003106         MOV     #128.,NEWCYL    ;SET NEWCYL TO 128
2776  026762  022737  000001  002302         CMP     #1,T.DRIVE      ;RL01?
2777  026770  001403                         BEQ     102$            ;YUP
2778  026772  012737  000400  003106         MOV     #256.,NEWCYL    ;SET FOR RL02
2779  027000  000137  026326          102$:  JMP     8$              ;DO SEEK LOOP
2780  027004  022737  000002  003236  24$:   CMP     #2,PASCNT       ;TEST IF PASS 2
2781  027012  001033                         BNE     28$             ;NO - SKIP
2782  027014  013737  002312  003106         MOV     NXTHL,NEWCYL    ;SET UP TO TIME 1 CYL SEEK AT INNER
2783  027022  004737  017326                 JSR     PC,XSEEK        ;  LIMIT
2784  027026  030052                         65$
2785  027030  012701  005670                 MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
2786  027034  004737  022222                 JSR     PC,RDYWAIT      ;WAIT FOR READY
2787  027040  030052                         65$
2788  027042  004737  022634                 JSR     PC,VERPOS       ;VERIFY POSITION
2789  027046  030052                         65$
2790  027050  012700  003144                 MOV     #OFIN,R0        ;SET POINTERS
2791  027054  012701  003146                 MOV     #OFINU,R1
2792  027060  012703  003160                 MOV     #ORIN,R3
2793  027064  012704  003162                 MOV     #ORINU,R4
2794  027070  013737  002306  00310'         MOV     HLMTW,NEWCYL    ;LOAD NEW CYLINDER
2795  027076  000137  026326                 JMP     8$              ;DO SEEK LOOP
2796  027102  022737  000003  003236  28$:   CMP     #3,PASCNT       ;TEST IF PASS 3
2797  027110  001040                         BNE     32$             ;NO - SKIP
2798  027112  005037  003106                 CLR     NEWCYL          ;ELSE SET UP TO TIME 85/170 CYL SEEK
2799  027116  004737  017326                 JSR     PC,XSEEK        ;  AT OUTER LIMIT
2800  027122  030052                         65$
2801  027124  012701  005670                 MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
2802  027130  004737  022222                 JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
2803  027134  030052                         65$
2804  027136  004737  022634                 JSR     PC,VERPOS       ;VERIFY POSITION
2805  027142  030052                         65$
2806  027144  012700  003200                 MOV     #HFOUT,R0       ;SET POINTERS
2807  027150  012701  003202                 MOV     #HFOUTU,R1
2808  027154  012703  003210                 MOV     #HROUT,R3
2809  027160  012704  003202                 MOV     #HFOUTU,R4
2810  027164  012737  000125  003106         MOV     #85.,NEWCYL     ;LOAD NEWCYL FOR 85 CYL SEEK
2811  027172  022737  000001  002302         CMP     #1,T.DRIVE      ;RL01?
2812  027200  001505                         BEQ     39$             ;YUP
2813  027202  012737  000252  003106         MOV     #170.,NEWCYL    ;NO - SET FOR RL02
2814  027210  000501                         BR      39$
2815  027212  022737  000004  003236  32$:   CMP     #4,PASCNT       ;TEST IF PASS 4
2816  027220  001041                         BNE     36$             ;NO - SKIP
2817  027222  012737  000252  003106         MOV     #170.,NEWCYL    ;ELSE SET UP TO TIME 85 CYL SEEK
2818  027230  022737  000001  002302         CMP     #1,T.DRIVE      ;RL01?
2819  027236  001403                         BEQ     321$            ;YES
```

```
 2820  027240  012737  000525  003106          MOV     #341.,NEWCYL    ;NO - SET FOR RL02
 2821  027246  004737  017326          321$:   JSR     PC,XSEEK        ; AT INNER LIMIT
 2822  027252  030052                          65$
 2823  027254  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 2824  027260  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR READY
 2825  027264  030052                          65$
 2826  027266  004737  022634                  JSR     PC,VERPOS       ;VERIFY POSITION
 2827  027272  030052                          65$
 2828  027274  012700  003174                  MOV     #HFIN,R0        ;SET POINTERS
 2829  027300  012701  003176                  MOV     #HFINU,R1
 2830  027304  012703  003204                  MOV     #HRIN,R3
 2831  027310  012704  003206                  MOV     #HRINU,R4
 2832  027314  013737  002306  003106          MOV     HLMTW,NEWCYL    ;SET NEWCYL TO 255/511 FOR 85/170 CYL SEEK
 2833  027322  000434                          BR      39$             ;DO TIMING LOOP
 2834  027324  022737  000005  003236  36$:    CMP     #5,PASCNT       ;TEST IF PASS 5
 2835  027332  001032                          BNE     40$             ;NO - SKIP
 2836  027334  005037  003106                  CLR     NEWCYL          ;ELSE SET UP TO TIME 256/512 CYL SEEK
 2837  027340  004737  017326                  JSR     PC,XSEEK        ;  OVER ALL SURFACE
 2838  027344  030052                          65$
 2839  027346  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 2840  027352  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
 2841  027356  030052                          65$
 2842  027360  004737  022634                  JSR     PC,VERPOS       ;VERIFY POSITION
 2843  027364  030052                          65$
 2844  027366  012700  003214                  MOV     #AFMID,R0       ;SET POINTERS
 2845  027372  012701  003216                  MOV     #AFMIDU,R1
 2846  027376  012703  003220                  MOV     #ARMID,R3
 2847  027402  012704  003222                  MOV     #ARMIDU,R4
 2848  027406  013737  002306  003106          MOV     HLMTW,NEWCYL    ;SET NEWCYL
 2849  027414  000137  026326          39$:    JMP     8$
 2850  027420                          40$:    PRINTF  #FMT1.1,#SKTMES,#VALDES
  (9)  027420  012746  007117                  MOV     #VALDES,-(SP)
  (8)  027424  012746  007063                  MOV     #SKTMES,-(SP)
  (7)  027430  012746  011144                  MOV     #FMT1.1,-(SP)
  (6)  027434  012746  000003                  MOV     #3,-(SP)
  (3)  027440  010600                          MOV     SP,R0
  (4)  027442  104417                          TRAP    C$PNTF
  (4)  027444  062706  000010                  ADD     #10,SP
 2851  027450                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11)  027450  005046                          CLR     -(SP)
 (11)  027452  153716  003037                  BISB    RLDRV+1,(SP)
 (10)  027456  012746  006142                  MOV     #DRVNAM,-(SP)
  (9)  027462  013746  003032                  MOV     RLBAS,-(SP)
  (8)  027466  012746  006131                  MOV     #BASADD,-(SP)
  (7)  027472  012746  011172                  MOV     #FMT5,-(SP)
  (6)  027476  012746  000005                  MOV     #5,-(SP)
  (3)  027502  010600                          MOV     SP,R0
  (4)  027504  104417                          TRAP    C$PNTF
  (4)  027506  062706  000014                  ADD     #14,SP
 2852  027512                                  PRINTF  #FMT18,#LABIN,#LABMID,#LABOUT,#LABEXP
 (11)  027512  012746  007176                  MOV     #LABEXP,-(SP)
 (10)  027516  012746  007170                  MOV     #LABOUT,-(SP)
  (9)  027522  012746  007161                  MOV     #LABMID,-(SP)
  (8)  027526  012746  007153                  MOV     #LABIN,-(SP)
  (7)  027532  012746  011564                  MOV     #FMT18,-(SP)
  (6)  027536  012746  000005                  MOV     #5,-(SP)
```

```
   (3)  027542  010600                    MOV     SP,R0
   (4)  027544  104417                    TRAP    C$PNTF
   (4)  027546  062706  000014            ADD     #14,SP
  2853  027552                            PRINTF  #FMT19,#LABOCF,OFIN,OFMID,OFOUT,EXOCYL
  (12)  027552  013746  003224            MOV     EXOCYL,-(SP)
  (11)  027556  013746  003154            MOV     OFOUT,-(SP)
  (10)  027562  013746  003150            MOV     OFMID,-(SP)
   (9)  027566  013746  003144            MOV     OFIN,-(SP)
   (8)  027572  012746  007207            MOV     #LABOCF,-(SP)
   (7)  027576  012746  011616            MOV     #FMT19,-(SP)
   (6)  027602  012746  000006            MOV     #6,-(SP)
   (3)  027606  010600                    MOV     SP,R0
   (4)  027610  104417                    TRAP    C$PNTF
   (4)  027612  062706  000016            ADD     #16,SP
  2854  027616                            PRINTF  #FMT19,#LABOCR,ORIN,ORMID,OROUT,EXOCYL
  (12)  027616  013746  003224            MOV     EXOCYL,-(SP)
  (11)  027622  013746  003170            MOV     OROUT,-(SP)
  (10)  027626  013746  003164            MOV     ORMID,-(SP)
   (9)  027632  013746  003160            MOV     ORIN,-(SP)
   (8)  027636  012746  007221            MOV     #LABOCR,-(SP)
   (7)  027642  012746  011616            MOV     #FMT19,-(SP)
   (6)  027646  012746  000006            MOV     #6,-(SP)
   (3)  027652  010600                    MOV     SP,R0
   (4)  027654  104417                    TRAP    C$PNTF
   (4)  027656  062706  000016            ADD     #16,SP
  2855  027662                            PRINTF  #FMT20,#LABHCF,HFIN,HFOUT,EXHCYL
  (11)  027662  013746  003226            MOV     EXHCYL,-(SP)
  (10)  027666  013746  003200            MOV     HFOUT,-(SP)
   (9)  027672  013746  003174            MOV     HFIN,-(SP)
   (8)  027676  012746  007233            MOV     #LABHCF,-(SP)
   (7)  027702  012746  011653            MOV     #FMT20,-(SP)
   (6)  027706  012746  000005            MOV     #5,-(SP)
   (3)  027712  010600                    MOV     SP,R0
   (4)  027714  104417                    TRAP    C$PNTF
   (4)  027716  062706  000014            ADD     #14,SP
  2856  027722                            PRINTF  #FMT20,#LABHCR,HRIN,HROUT,EXHCYL
  (11)  027722  013746  003226            MOV     EXHCYL,-(SP)
  (10)  027726  013746  003210            MOV     HROUT,-(SP)
   (9)  027732  013746  003204            MOV     HRIN,-(SP)
   (8)  027736  012746  007247            MOV     #LABHCR,-(SP)
   (7)  027742  012746  011653            MOV     #FMT20,-(SP)
   (6)  027746  012746  000005            MOV     #5,-(SP)
   (3)  027752  010600                    MOV     SP,R0
   (4)  027754  104417                    TRAP    C$PNTF
   (4)  027756  062706  000014            ADD     #14,SP
  2857  027762                            PRINTF  #FMT21,#LABACF,AFMID,EXACYL
  (10)  027762  013746  003230            MOV     EXACYL,-(SP)
   (9)  027766  013746  003214            MOV     AFMID,-(SP)
   (8)  027772  012746  007263            MOV     #LABACF,-(SP)
   (7)  027776  012746  011703            MOV     #FMT21,-(SP)
   (6)  030002  012746  000004            MOV     #4,-(SP)
   (3)  030006  010600                    MOV     SP,R0
   (4)  030010  104417                    TRAP    C$PNTF
   (4)  030012  062706  000012            ADD     #12,SP
  2858  030016                            PRINTF  #FMT21,#LABACR,ARMID,EXACYL
  (10)  030016  013746  003230            MOV     EXACYL,-(SP)
```

```
   (9)  030022  013746  003220                    MOV     ARMID,-(SP)
   (8)  030026  012746  007277                    MOV     #LABACR,-(SP)
   (7)  030032  012746  011703                    MOV     #FMT21,-(SP)
   (6)  030036  012746  00000'                    MOV     #4,-(SP)
   (3)  030042  010600                            MOV     SP,R0
   (4)  030044  104417                            TRAP    C$PNTF
   (4)  030046  062706  000012                    ADD     #12,SP
2859    030052                          65$:
2860    030052                          ENDTST
   (3)  030052                          L10023:
   (3)  030052  104401                            TRAP    C$ETST
```

```
                                        .SBTTL  *TEST 2          **BASIC READ DATA (BAD SECTOR FILE)
030054                                  BGNTST                   ;TEST 2
(3) 030054                                                                                  T2::
030054  012737  006676  003016                  MOV     #P2T13E,ERHEAD  ;SET ERROR HEADER
030062  004737  016362                          JSR     PC,TSTINT       ;INITIALIZE TEST
030066  004737  016400                          JSR     PC,GSTATR       ;CLEAR DRIVE
030072  030542                                  65$
030074  012737  000001  003116                  MOV     #1,DESHD        ;SET TO HEAD 1
030102  032737  010000  013722                  BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPEC'D
030110  001405                                  BEQ     2$              ;NO - SKIP
030112  005737  013730                          TST     HEADW           ;TEST IF HEAD 0
030116  001002                                  BNE     2$              ;NO - SKIP
030120                                          EXIT    TST             ;ELSE EXIT TEST
(3) 030120  104432                              TRAP    C$EXIT
(3) 030122  000446                              .WORD   L10024-.
030124  013737  002306  003106  2$:             MOV     HLMTW,NEWCYL    ;POSITION HEADS AT 255
030132  004737  017326                          JSR     PC,XSEEK        ;DO SEEK
030136  030542                                  65$
030140  012701  005670                          MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
030144  004737  022222                          JSR     PC,RDYWAIT      ;WAIT FOR INTERRUPT
030150  030542                                  65$
030152  004737  022634                          JSR     PC,VERPOS       ;VERIFY POSITION
030156  030542                                  65$
030160  005037  003120                          CLR     DESSEC          ;SET FOR SECTOR 0
030164  012737  003676  003134                  MOV     #FBSFIL,TEMP5   ;SET TEMP STORAGE FOR FACTORY BS FILE
030172  012737  000020  003136                  MOV     #16.,TEMP6      ;SET MAX SECTOR COUNT
030200  112737  000001  003451                  MOVB    #1,NOERCT       ;SET FOR NO ERROR COUNTING
030206  105037  003450                          CLRB    LOCERR          ;CLEAR LOCAL ERROR COUNTER
030212  005037  003130          4$:             CLR     TEMP3           ;CLEAR ONES DETECTED FLAG
030216  013701  003134                          MOV     TEMP5,R1        ;INIT POINTERS
030222  013700  003136                          MOV     TEMP6,R0
030226  012703  004072                          MOV     #IBUFF,R3
030232  012737  000002  003022                  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
030240  004737  024014                          JSR     PC,XREAD        ;DO READ
030244  030416                                  39$
030246  005723                                  TST     (R3)+           ;TEST IF WORD 0 NOT NEG
030250  100516                                  BMI     45$             ;YES, BAD FMT ERROR
030252  005723                                  TST     (R3)+           ;ELSE TEST WORD 1 NOT NEG
030254  100514                                  BMI     45$             ;YES - BAD FMT ERROR REPORT
030256  005723                          7$:     TST     (R3)+           ;TEST WORD 2 IS 0
030260  001112                                  BNE     45$             ;NO - SKIP TO FMT ERROR RPT
030262  005723                                  TST     (R3)+           ;TEST WORD 3 IS 0
030264  001110                                  BNE     45$             ;NO - SKIP TO FMT ERROR RPT
030266  021327  177777          8$:             CMP     (R3),#-1        ;TEST IF NEXT WORD IS ALL 1'S
030272  001004                                  BNE     10$             ;NO  - SKIP
030274  012737  000001  003130                  MOV     #1,TEMP3        ;ELSE SET 1'S DETECTED FLAG
030302  000403                                  BR      11$             ;SKIP
030304  005737  003130          10$:            TST     TEMP3           ;TEST IF ONES HAVE BEEN DETECTED
030310  001076                                  BNE     45$             ;YES - SKIP TO FMT ERROR RPT
030312  012311                          11$:    MOV     (R3)+,(R1)      ;STORE CYLINDER WORD
030314  012705  000007                          MOV     #7,R5           ;ALIGN IT TO LOOK LIKE HEADER
030320  006311                          12$:    ASL     (R1)
030322  005305                                  DEC     R5
030324  001375                                  BNE     12$
030326  032713  000400                          BIT     #BIT8,(R3)      ;TEST IF HEAD 1
030332  001402                                  BEQ     15$             ;NO - SKIP
```

```
 2915  030334  052711  000100                  BIS     #BIT6,(R1)          ;INSERT HEAD BIT
 2916  030340  042713  177400          15$:    BIC     #177400,(R3)        ;CLEAR ALL BUT SECTOR
 2917  030344  052321                          BIS     (R3)+,(R1)+         ;INSERT SECTOR NUMBER
 2918  030346  020327  004472                  CMP     R3,#IBUFF+256.      ;CHECK IF IBUFF EMPTY
 2919  030352  001345                          BNE     8$                  ;NO GET NEXT CYLINDER
 2920  030354  005737  003130                  TST     TEMP3               ;ELSE TEST IF 1'S DETECTED
 2921  030360  001461                          BEQ     48$                 ;TO MANY ERRORS - REPORT
 2922  030362  022737  000044  003136          CMP     #36.,TEMP6          ;CHECK IF SOFTWARE BAD READ
 2923  030370  001464                          BEQ     65$                 ;YES - SKIP
 2924  030372  012737  003502  003134  37$:    MOV     #SBSFIL,TEMP5       ;ELSE CHANGE POINTERS
 2925  030400  012737  000044  003136          MOV     #36.,TEMP6          ;       MAX SECTOR NUMBER
 2926  030406  012737  000024  003120          MOV     #20.,DESSEC         ;       SECTOR NUMBER START
 2927  030414  000676                          BR      4$                  ;DO READ
 2928  030416  005237  003450          39$:    INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
 2929  030422  012777  177777  152504  40$:    MOV     #-1,@TEMP5          ;MOV    1'S INTO FILE STORAGE
 2930  030430                                  INLOOP                      ;CHECK IF IN ERROR LOOP
   (3) 030430  104420                          TRAP    C$INLP
 2931  030432                                  BCOMPLETE       4$                  ;YES - GO DO READ
   (2) 030432  103667                          BCS     4$
 2932  030434  023737  003120  003136  41$:    CMP     DESSEC,TEMP6        ;CHECK IF ALL SECTORS READ
 2933  030442  001015                          BNE     43$                 ;NO - SKIP
 2934  030444  012703  006033                  MOV     #MBADSF,R3          ;SET RESULT MESSAGE POINTER
 2935  030450  005237  003450                  INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
 2936  030454                                  ERRHRD  1301.,,ERR1
   (4) 030454  104456                          TRAP    C$ERHRD
   (5) 030456  002425                          .WORD   1301
   (5) 030460  000000                          .WORD   0
   (5) 030462  012070                          .WORD   ERR1
 2937  030464  022737  003502  003134          CMP     #SBSFIL,TEMP5       ;TEST IF SOFTWARE FILES CHECKED
 2938  030472  001423                          BEQ     65$                 ;YES - EXIT
 2939  030474  000736                          BR      37$                 ;ELSE GO CHECK SOFTWARE FILES
 2940  030476  062737  000004  003120  43$:    ADD     #4,DESSEC           ;BUMP TO NEXT SECTOR
 2941  030504  000642                          BR      4$                  ;GO DO READ
 2942  030506  012703  006063          45$:    MOV     #MFMTER,R3          ;SET RESULT MESSAGE POINTER
 2943  030512                                  ERRHRD  1302.,,ERR1
   (4) 030512  104456                          TRAP    C$ERHRD
   (5) 030514  002426                          .WORD   1302
   (5) 030516  000000                          .WORD   0
   (5) 030520  012070                          .WORD   ERR1
 2944  030522  000735                          BR      39$                 ;GO CHECK FOR LOOP
 2945  030524  012703  006110          48$:    MOV     #MTMBS,R3           ;SET RESULT MESSAGE PTR
 2946  030530                                  ERRHRD  1303.,,ERR1
   (4) 030530  104456                          TRAP    C$ERHRD
   (5) 030532  002427                          .WORD   1303
   (5) 030534  000000                          .WORD   0
   (5) 030536  012070                          .WORD   ERR1
 2947  030540  000730                          BR      40$                 ;GO CHECK FOR LOOP
 2948  030542  012737  000002  003022  65$:    MOV     #2,ERRSWI           ;INIT ERROR SWITCH
 2949  030550  012737  000001  003500          MOV     #1,BSFVAL           ;SET BAD SECTOR FILES VALID FLAG
 2950  030556  105737  003450                  TSTB    LOCERR              ;TEST IF LOCAL ERRORS
 2951  030562  001402                          BEQ     66$                 ;NO - SKIP
 2952  030564  005237  003244                  INC     ERRCNT              ;ELSE BUMP ERROR COUNT
 2953  030570                          66$:
 2954  030570                          ENDTST
   (3) 030570                          L10024:
   (3) 030570  104401                          TRAP    C$ETST
```

```
2956
2957  030572                                  .SBTTL  *TEST 3          **WRITE/READ DATA (PART 1)
2957  030572                                  BGNTST          ;TEST 3
 (3)  030572                                                                                  T3::
2958  030572  012737  006712  003016                  MOV     #P2T14E,ERHEAD  ;SET ERROR HEADER
2959  030600  004737  021030                          JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
2960  030604  004737  016362                          JSR     PC,TSTINT       ;INITIALIZE TEST
2961  030610  004737  016400                          JSR     PC,GSTATR       ;CLEAR DRIVE
2962  030614  031004                                  T3065$
2963  030616  004737  020720                          JSR     PC,CHOSHD *     ;GO CHOSE HEAD
2964  030622  005037  003120                          CLR     DESSEC          ;       SECTOR 0
2965  030626  005037  003106                          CLR     NEWCYL          ;       CYLINDER 0
2966  030632  005037  030676                          CLR     T310$           ;CLEAR PATTERN SELECT
2967  030636  004737  017326          T306$:  JSR     PC,XSEEK        ;POSITION HEADS
2968  030642  031004                                  T3065$
2969  030644  012701  005670                          MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
2970  030650  004737  022222                          JSR     PC,RDYWAIT      ;WAIT FOR READY
2971  030654  031004                                  T3065$
2972  030656  004737  022634                          JSR     PC,VERPOS       ;VERIFY POSITION
2973  030662  031004                                  T3065$
2974  030664  005037  030676                          CLR     T~10$           ;CLEAR PATTERN SELECTOR
2975  030670                          T307$:
2976  030670                          BGNSUB
 (3)  030670                                                                                  T3.1:
 (3)  030670  104402                                  TRAP    C$BSUB
2977  030672  004537  023324                          JSR     R5,DATGEN       ;GENERATE DATA
2978  030676  000000                  T310$:  .WORD   0               ;PATTERN SELECT WORD
2979  030700  004737  023754                          JSR     PC,XWRITE       ;DO WRITE DATA
2980  030704  030722                                  60$
2981  030706  004737  024014                          JSR     PC,XREAD        ;DO READ DATA
2982  030712  030722                                  60$
2983  030714  004737  023464                          JSR     PC,DATCOM       ;COMPARE DATA
2984  030720  030722                                  60$
2985  030722  012737  000002  003022  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
2986  030730                          ENDSUB
 (3)  030730                          L10026:
 (3)  030730  104403                                  TRAP    C$ESUB
2987  030732                                  ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  030732  104410                                  TRAP    C$ESCAPE
 (3)  030734  000050                                  .WORD   L10025-.
2988  030736  022737  000010  030676                  CMP     #8.,T310$       ;WAS DATA PAT 8 USED?
2989  030744  001403                                  BEQ     10$             ;YES - SKIP
2990  030746  005237  030676                          INC     T310$           ;ELSE BUMP TO NEXT PATTERN
2991  030752  000746                                  BR      T307$           ;DO TEST WITH NEW PATTERN
2992  030754  004737  020744          10$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
2993  030760  031004                                  T3065$                  ;ABORT RETURN
2994  030762  005037  030676                          CLR     T310$           ;SET PATTERN SELECT TO 0
2995  030766  004737  024512          11$:    JSR     PC,BSCHK        ;CHECK IF SECTOR BAD
2996  030772  030776                                  13$             ;YES RETURN - SKIP TO 13$
2997  030774  000720                                  BR      T306$           ;NO RETURN - DO TEST THIS SECTOR
2998  030776  005237  003106          13$:    INC     NEWCYL          ;BUMP TO NEXT CYLINDER
2999  031002  000771                                  BR      11$             ;CHECK IF THIS ONE BAD
3000  031004                          T3065$:
3001  031004                          ENDTST
 (3)  031004                          L10025:
 (3)  031004  104401                                  TRAP    C$ETST
3002
```

```
3004                                        .SBTTL  *TEST 4          **ROTATIONAL TIMING
3005  031006                                BGNTST          ;TEST 4
 (3)  031006                                                                        T4::
3006  031006  012737  006733  003016              MOV     #P2T15E,ERHEAD  ;SET ERROR HEADER
3007                                        ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3008  031014  005737  003474                      TST     CLKFLG          ;P-CLOCK?
3009  031020  001014                              BNE     3$              ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
3010  031022                                      PRINTF  #FMT9,#NOTST4   ;ELSE, PRINT MSG. 'TEST 4 CANNOT BE PERFORMED...
 (8)  031022  012746  010036                      MOV     #NOTST4,-(SP)
 (7)  031026  012746  011356                      MOV     #FMT9,-(SP)
 (6)  031032  012746  000002                      MOV     #2,-(SP)
 (3)  031036  010600                              MOV     SP,R0
 (4)  031040  104417                              TRAP    C$PNTF
 (4)  031042  062706  000006                      ADD     #6,SP
3011                                                                    ;/P-CLOCK IS NOT AVAILABLE''
3012  031046                                      EXIT    TST
 (3)  031046  104432                              TRAP    C$EXIT
 (3)  031050  000542                              .WORD   L10027-.
3013  031052  005003                      3$:     CLR     R3              ;CLEAR FOR TIMING STORAGE
3014  031054  005004                              CLR     R4
3015  031056  004737  016362                      JSR     PC,TSTINT       ;INITIALIZE TEST
3016  031062  004737  016400                      JSR     PC,GSTATR       ;CLEAR DRIVE
3017  031066  031604                              60$
3018  031070  004537  023324                      JSR     R5,DATGEN       ;GENERATE DATA
3019  031074  000000                              0                       ;PATTERN 0
3020  031076  005037  003120                      CLR     DESSEC          ;CLEAR TO SECTOR 0
3021  031102  004737  020720                      JSR     PC,CHOSHD       ;GO SELECT HEAD
3022  031106  013737  013724  003106              MOV     LOLIMW,NEWCYL   ;SET FOR CYLINDER
3023  031114  004737  017326                      JSR     PC,XSEEK        ;DO SEEK
3024  031120  031604                              60$
3025  031122  012701  005670                      MOV     #3000.,R1       ;SET WAIT FOR 300 MS
3026  031126  004737  022222                      JSR     PC,RDYWAIT      ;WAIT FOR READY
3027  031132  031604                              60$
3028  031134  004737  022634                      JSR     PC,VERPOS       ;VERIFY POSITION
3029  031140  031604                              60$
3030  031142  012701  000100                      MOV     #64.,R1 ;SET LOOP COUNTER
3031  031146  012705  003046              5$:     MOV     #L.MP,R5        ;SET A POINTER
3032  031152  004737  023744                      JSR     PC,XWRITT       ;DO FIRST WRITE
3033  031156  031604                              60$
3034  031160  011562  000006                      MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS
3035  031164  014562  000004                      MOV     -(R5),RLDA(R2)
3036  031170  014562  000002                      MOV     -(R5),RLBA(R2)
3037  031174  014562  000000                      MOV     -(R5),RLCS(R2)
3038  031200                                      WAITUS  #3000.
3039  031212  005737  003012                      TST     DONE            ;TEST IF INTERRUPT
3040  031216  001011                              BNE     6$              ;YES - SKIP
3041  031220  004737  016224                      JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
3042  031224  012603                              MOV     (SP)+,R3        ;GET MESSAGE POINTER
3043  031226                                      ERRHRD  1501.,,ERR1
 (4)  031226  104456                              TRAP    C$ERHRD
 (5)  031230  002735                              .WORD   1501
 (5)  031232  000000                              .WORD   0
 (5)  031234  012070                              .WORD   ERR1
3044  031236  000137  031604                      JMP     60$
3045  031242  005737  003050              6$:     TST     T.CS            ;TEST IF ANY ERRORS
3046  031246  100006                              BPL     4$              ;NO - SKIP
```

```
3047  031250                                    ERRHRD  1502.,,ERR6
 (4)  031250  104456                            TRAP    C$ERHRD
 (5)  031252  002736                            .WORD   1502
 (5)  031254  000000                            .WORD   0
 (5)  031256  012372                            .WORD   ERR6
3048  031260  000137  031604                    JMP     60$
3049  031264  012705  003046          4$:       MOV     #L.MP,R5        ;SET POINTER TO RL LOAD REGS
3050  031270  005037  003012                    CLR     DONE            ;CLEAR INTERRUPT INDICATOR
3051  031274                                    STCLK                   ;START P-CLOCK TO INITIATE MEASUREMENT
3052                                                                    ;/OF TIME INTERVAL
3053  031312  011562  000006                    MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS FOR 2ND WRITE
3054  031316  014562  000004                    MOV     -(R5),RLDA(R2)
3055  031322  014562  000002                    MOV     -(R5),RLBA(R2)
3056  031326  014562  000000                    MOV     -(R5),RLCS(R2)
3057  031332                                    WAITUS  #3000.          ;WAIT FOR INTERRUPT
3058  031344                                    GETTIM  R0              ;GET ELAPSED TIME
3059  031354  005737  003012                    TST     DONE            ;TEST IF INTERRUPT OCCURRED
3060  031360  001010                            BNE     7$              ;YES - SKIP
3061  031362  004737  016224                    JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
3062  031366  012603                            MOV     (SP)+,R3        ;GET MESSAGE POINTER
3063  031370                                    ERRHRD  1503.,,ERR1     ;REPORT
 (4)  031370  104456                            TRAP    C$ERHRD
 (5)  031372  002737                            .WORD   1503
 (5)  031374  000000                            .WORD   0
 (5)  031376  012070                            .WORD   ERR1
3064  031400  000501                            BR      60$
3065  031402  005737  003050          7$:       TST     T.CS            ;TEST IF ANY ERROR
3066  031406  100005                            BPL     8$              ;NO - SKIP
3067  031410                                    ERRHRD  1504.,,ERR6     ;REPORT ERRORS
 (4)  031410  104456                            TRAP    C$ERHRD
 (5)  031412  002740                            .WORD   1504
 (5)  031414  000000                            .WORD   0
 (5)  031416  012372                            .WORD   ERR6
3068  031420  000471                            BR      60$
3069  031422  060003                8$:         ADD     R0,R3           ;ADD IN TIME USED
3070  031424  005504                            ADC     R4              ;DOUBLE PRECISION
3071  031426  005301                            DEC     R1              ;DEC LOOP COUNTER
3072  031430  001246                            BNE     5$              ;LOOP UNTIL 0
3073  031432  012701  000006                    MOV     #6,R1           ;SET DIVIDE COUNT
3074  031436  000241                10$:        CLC                     ;CLEAR CARRY FOR DIVIDE
3075  031440  006004                            ROR     R4              ;DIVIDE SUM BY 100(8)
3076  031442  006003                            ROR     R3
3077  031444  005301                            DEC     R1              ;DEC DIVIDE COUNT
3078  031446  001373                            BNE     10$             ;LOOP UNTIL DONE
3079  031450                                    PRINTF  #FMT1.1,#SRTMES,#VALDES
 (9)  031450  012746  007117                    MOV     #VALDES,-(SP)
 (8)  031454  012746  007075                    MOV     #SRTMES,-(SP)
 (7)  031460  012746  011144                    MOV     #FMT1.1,-(SP)
 (6)  031464  012746  000003                    MOV     #3,-(SP)
 (3)  031470  010600                            MOV     SP,R0
 (4)  031472  104417                            TRAP    C$PNTF
 (4)  031474  062706  000010                    ADD     #10,SP
3080  031500                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 031500  005046                            CLR     -(SP)
 (11) 031502  153716  003037                    BISB    RLDRV+1,(SP)
 (10) 031506  012746  006142                    MOV     #DRVNAM,-(SP)
```

```
  (9)  031512  013746  003032                    MOV     RLBAS,-(SP)
  (8)  031516  012746  006131                    MOV     #BASADD,-(SP)
  (7)  031522  012746  011172                    MOV     #FMT5,-(SP)
  (6)  031526  012746  000005                    MOV     #5,-(SP)
  (3)  031532  010600                            MOV     SP,R0
  (4)  031534  104417                            TRAP    C$PNTF
  (4)  031536  062706  000014                    ADD     #14,SP
3081   031542                                    PRINTF  #FMT26,#RESE3,R3,#RESE4,#MAPROX,EXROT
 (12)  031542  013746  003232                    MOV     EXROT,-(SP)
 (11)  031546  012746  007143                    MOV     #MAPROX,-(SP)
 (10)  031552  012746  010731                    MOV     #RESE4,-(SP)
  (9)  031556  010346                            MOV     R3,-(SP)
  (8)  031560  012746  010725                    MOV     #RESE3,-(SP)
  (7)  031564  012746  012013                    MOV     #FMT26,-(SP)
  (6)  031570  012746  000006                    MOV     #6,-(SP)
  (3)  031574  010600                            MOV     SP,R0
  (4)  031576  104417                            TRAP    C$PNTF
  (4)  031600  062706  000016                    ADD     #16,SP
3082   031604  012737  000002  003022  60$:      MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
3083   031612                    ENDTST
  (3)  031612                    L10027:
  (3)  031612  104401                            TRAP    C$ETST
3084
```

```
3086                                        .SBTTL  *TEST 5          **WRITE/READ DATA (PART 2)
3087   031614                               BGNTST                   ;TEST 5
 (3)   031614                                                                              T5::
3088   031614  012737  006756  003016              MOV     #P2T16E,ERHEAD  ;SET ERROR HEADER
3089   031622  004737  021030                      JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
3090   031626  004737  016362                      JSR     PC,TSTINT       ;INITIALIZE TEST
3091   031632  004737  016400                      JSR     PC,GSTATR       ;CLEAR DRIVE
3092   031636  032722                              T3165$
3093   031640  005037  003236                      CLR     PASCNT          ;CLEAR PASS TO 0
3094   031644  012705  177776                      MOV     #-2,R5          ;SET
3095   031650  005737  003444                      TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
3096   031654  001006                              BNE     1$              ;NO - SKIP
3097   031656  032737  000001  013722              BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
3098   031664  001002                              BNE     1$              ;YES - SKIP
3099   031666  012705  177760                      MOV     #-16.,R5                ;ELSE SET PEOPLE TO NEG 8
3100   031672                               1$:
3101   031672  012701  002510                      MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
3102   031676  012737  000010  002304              MOV     #10,JJJ         ;SET CLEAR COUNT
3103   031704  013721  013724               2$:    MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LO LIMIT
3104   031710  005337  002304                      DEC     JJJ             ;DEC COUNT
3105   031714  001373                              BNE     2$              ;LOOP UNTIL 0
3106   031716  013737  013726  002514              MOV     HILIMW,T33TBL+4 ;INSERT HILIMIT
3107   031724  013737  013726  002516              MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
3108   031732  013737  013726  002520              MOV     HILIMW,T33TBL+10
3109   031740  062705  000002               T3100$: ADD     #2,R5   ;BUMP R5 BY 2
3110   031744  032737  000001  013722              BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
3111   031752  001031                              BNE     5$              ;YES - SKIP
3112   031754  005737  003444                      TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
3113   031760  001002                              BNE     3$              ;NO - SKIP
3114   031762  062705  000016                      ADD     #16,R5          ;ELSE BUMP CYLINDER POINTER BY 7
3115   031766  022737  000001  002302       3$:    CMP     #1,T.DRIVE      ;RL01 OR RL02?THAT IS THE Q
3116   031774  001404                              BEQ     44$             ;ANS IS RL01
3117   031776  020527  000244                      CMP     R5,#164.
3118   032002  103013                              BHIS    4$
3119   032004  000403                              BR      69$             ;TEST PAST TABLE-YES EXIT
3120   032006  020527  000122               44$:   CMP     R5,#82.
3121   032012  103007                              BHIS    4$              ;TES PAST THE TABLE
3122
3123   032014  016537  002610  002304       69$:   MOV     CYLTBL(R5),JJJ  ;GET NEXT TABLE ENTRY
3124   032022  043737  002310  002304              BIC     CLRBYT,JJJ      ;CLEAR UPPER BYTE
3125   032030  001007                              BNE     8$
3126   032032  000137  032722               4$:    JMP     T3165$          ;EXIT TEST
3127   032036  023705  013726               5$:    CMP     HILIMW,R5       ;TEST IF ALL CYLINDERS USED
3128   032042  001773                              BEQ     4$              ;YES - EXIT TEST
3129   032044  010537  002304                      MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
3130   032050  023737  002304  013724       8$:    CMP     JJJ,LOLIMW      ;CHECK IF LOWER THAN LOLIMIT
3131   032056  103730                              BLO     T3100$          ;YES - SKIP
3132   032060  023737  002304  013726              CMP     JJJ,HILIMW      ;CHECK IF HIGHER THAN HILIMIT
3133   032066  101324                              BHI     T3100$          ;YES - SKIP
3134   032070  012703  002550                      MOV     #TBT,R3
3135   032074  013713  002304                      MOV     JJJ,(R3)
3136   032100  013763  002304  000002              MOV     JJJ,2(R3)
3137   032106  013763  002304  000004              MOV     JJJ,4(R3)
3138   032114  013763  002304  000006              MOV     JJJ,6(R3)
3139   032122  013763  002304  000010              MOV     JJJ,10(R3)
3140   032130  013763  002304  000012              MOV     JJJ,12(R3)
```

```
  3141  032136  010337  003030                MOV     R3,TBLSTR       ;STORE TABLE ADDRESS
  3142  032142  004737  020720                JSR     PC,CHOSHD       ;GO CHOSE HEAD
  3143
  3144  032146                        T3101$:
  3145  032146                        BGNSUB
   (3)  032146                                                                T5.1:
   (3)  032146  104402                TRAP    C$BSUB
  3146  032150  042737  003760  003010        BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
  3147  032156  005737  003236                TST     PASCNT          ;TEST IF PASS 0
  3148  032162  001414                        BEQ     11$             ;YES - SKIP
  3149  032164  023727  003236  000003        CMP     PASCNT,#3       ;TEST IF PASS 3
  3150  032172  001404                        BEQ     10$             ;YES - SKIP
  3151  032174  002407                        BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
  3152  032176  012737  000003  003236        MOV     #3,PASCNT       ;ELSE SET TO 3
  3153  032204  052737  000020  003010  10$:   BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
  3154  032212  000405                        BR      12$             ;SKIP
  3155  032214  005037  003236          11$:  CLR     PASCNT          ;SET PASS COUNT TO 0
  3156  032220  052737  000040  003010        BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
  3157  032226  012737  000003  003026  12$:   MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
  3158  032234  013703  003030                MOV     TBLSTR,R3       ;GET STORED TABLE ADDRESS
  3159  032240  012701  002510                MOV     #T33TBL,R1
  3160  032244  012703  002550                MOV     #TBT,R3
  3161  032250  005037  003120          15$:  CLR     DESSEC          ;CLEAR TO SECTOR 0
  3162  032254  012137  003106                MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
  3163  032260  004737  017326                JSR     PC,XSEEK        ;DO SEEK
  3164  032264  032630                        60$
  3165  032266  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
  3166  032272  004737  022222                JSR     PC,RDYWAIT      ;WAIT FOR READY
  3167  032276  032630                        60$
  3168  032300  012337  003106                MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
  3169  032304  004737  017326                JSR     PC,XSEEK        ;DO SEEK
  3170  032310  032630                        60$
  3171  032312  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
  3172  032316  004737  022222                JSR     PC,RDYWAIT      ;WAIT FOR READY
  3173  032322  032630                        60$
  3174  032324  004737  022634                JSR     PC,VERPOS       ;VERIFY POSITION
  3175  032330  032630                        60$
  3176  032332  004737  024512          16$:  JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
  3177  032336  032470                        32$                     ;"YES" RETURN
  3178  032340  013737  003120  032360        MOV     DESSEC,25$      ;SET DATA PATTERN = TO SECTOR NUMBER
  3179  032346  042737  177770  032360        BIC     #177770,25$     ;CLEAR ALL BUT LSD
  3180  032354  004537  023324                JSR     R5,DATGEN       ;GO GENERATE DATA
  3181  032360  000000                  25$:  .WORD   0
  3182  032362  032737  000001  003026        BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
  3183  032370  001425                        BEQ     29$             ;NO - SKIP
  3184  032372  004737  023754                JSR     PC,XWRITE       ;DO WRITE
  3185  032376  032630                        60$
  3186  032400  005237  003120                INC     DESSEC          ;INC SECTOR
  3187  032404  022737  000050  003120        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
  3188  032412  001347                        BNE     16$             ;NO - SKIP
  3189  032414  042737  000060  003010        BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
  3190  032422  042737  000001  003026        BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
  3191  032430  052737  000100  003010        BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
  3192  032436  005037  003120                CLR     DESSEC          ;CLEAR TO SECTOR 0
  3193  032442  000733                        BR      16$             ;SKIP
  3194  032444  032737  000002  003026  29$:  BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
```

```
3195  032452  001414                          BEQ     33$             ;NO - SKIP
3196  032454  004737  024014          31$:    JSR     PC,XREAD        ;ELSE DO READ
3197  032460  032630                          60$
3198  032462  004737  023464                  JSR     PC,DATCOM       ;COMPARE DATA
3199  032466  032630                          60$
3200  032470  005237  003120          32$:    INC     DESSEC          ;BUMP SECTOR
3201  032474  022737  000050  003120          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3202  032502  001313                          BNE     16$             ;NO - LOOP
3203  032504  005037  003120          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
3204  032510  005037  003026                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
3205  032514  005237  003236                  INC     PASCNT          ;BUMP PASS COUNT
3206  032520  042737  003760  003010          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
3207  032526  023727  003236  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
3208  032534  001435                          BEQ     60$             ;YES - SKIP
3209  032536  023727  003236  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
3210  032544  001431                          BEQ     60$             ;YES - SKIP
3211  032546  012737  000002  003026          MOV     #BIT1,WRTSWI    ;SET READ REQUIRED BIT
3212  032554  023727  003236  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
3213  032562  001415                          BEQ     40$             ;YES - SKIP
3214  032564  023727  003236  000005          CMP     PASCNT,#5       ;TEST IF PASS 4
3215  032572  001411                          BEQ     40$             ;YES - SKIP
3216  032574  000404                          BR      39$             ;SKIP
3217  032576  052737  002000  003010  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
3218  032604  000407                          BR      36$             ;GO DO NEXT PASS
3219  032606  052737  000020  003010  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
3220  032614  000403                          BR      36$             ;SKIP
3221  032616  052737  000040  003010  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
3222  032624  000137  032250          36$:    JMP     15$             ;GO DO NEXT PASS
3223  032630  012737  000002  003022  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3224  032636                          ENDSUB
 (3)  032636                          L10031:
 (3)  032636  104403                          TRAP    C$ESUB
3225  032640                                  ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  032640  104410                          TRAP    C$ESCAPE
 (3)  032642  000060                          .WORD   L10030-.
3226  032644  012737  000003  003026          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
3227  032652  023727  003236  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3228  032660  001004                          BNE     45$             ;NO - SKIP
3229  032662  012737  002516  003030          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
3230  032670  000410                          BR      48$             ;GO START PASS 4
3231  032672  005037  003236          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
3232  032676  004737  020744                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
3233  032702  031740                          T3100$                  ;ABORT RETURN
3234  032704  012737  002510  003030          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
3235  032712  062703  000006          48$:    ADD     #6,R3
3236  032716  000137  032146                  JMP     T3101$
3237  032722                          T3165$:
3238  032722                          ENDTS1
 (3)  032722                          L10030:
 (3)  032722  104401                          TRAP    C$ETST
```

```
3240                                        .SBTTL  *TEST 6          **WRITE LOCK ERROR AND DATA PROTECTION
3241  032724                                BGNTST                   ;TEST 6
 (3)  032724                                                                                    T6::
3242  032724  005737  003444                      TST     PASNUM           ;TEST IF FIRST PASS
3243  032730  001003                              BNE     2$               ;NO - SKIP
3244  032732  005737  013722                      TST     MISWIW           ;TEST IF RUN MANUAL INTERVENTION
3245  032736  100402                              BMI     3$               ;YES - SKIP
3246  032740  000137  033740              2$:     JMP     T3265$           ;EXIT TST
3247  032744                              3$:
3248  032744                                      BGNSUB
 (3)  032744                                                                                    T6.1:
 (3)  032744  104402                              TRAP    C$BSUB
3249  032746  012737  006777  003016              MOV     #P2T17E,ERHEAD   ;SET ERROR HEADER
3250  032754  004737  016362                      JSR     PC,TSTINT        ;INITIALIZE TEST
3251  032760  004737  016400                      JSR     PC,GSTATR        ;CLEAR DRIVE
3252  032764  033606                              60$
3253  032766  005037  003116                      CLR     DESHD            ;SET TO HEAD 0
3254  032772  005037  003120                      CLR     DESSEC           ;SET TO SECTOR 0
3255  032776  005037  003106                      CLR     NEWCYL           ;CLEAR TO CYLINDER 0
3256  033002  004737  017326                      JSR     PC,XSEEK         ;DO SEEK
3257  033006  033606                              60$
3258  033010  012701  013560                      MOV     #6000.,R1        ;INITIALIZE WAIT COUNT
3259  033014  004737  022222                      JSR     PC,RDYWAIT       ;WAIT FOR READY
3260  033020  033606                              60$
3261  033022  004737  022634                      JSR     PC,VERPOS        ;VERIFY POSITION
3262  033026  033606                              60$
3263  033030  032737  020000  003056              BIT     #WLSTAT,T.MP     ;TEST IF WRITE LOCK SET
3264  033036  001116                              BNE     7$               ;YES - SKIP
3265  033040  004537  023324                      JSR     R5,DATGEN        ;GENERATE DATA
3266  033044  000007                              7                        ;PATTERN 7
3267  033046  004737  023754                      JSR     PC,XWRITE        ;WRITE DATA
3268  033052  033606                              60$
3269  033054  004737  024014                      JSR     PC,XREAD         ;READ DATA
3270  033060  033606                              60$
3271  033062  004737  023464                      JSR     PC,DATCOM        ;CHECK DATA
3272  033066  033606                              60$
3273  033070                                      PRINTF  #FMTOP1,#OPR004,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST SET WR
 (13) 033070  005046                              CLR     -(SP)
 (13) 033072  153716  003037                      BISB    RLDRV+1,(SP)
 (12) 033076  012746  006142                      MOV     #DRVNAM,-(SP)
 (11) 033102  013746  003032                      MOV     RLBAS,-(SP)
 (10) 033106  012746  006131                      MOV     #BASADD,-(SP)
 (9)  033112  012746  007366                      MOV     #OPR1A,-(SP)
 (8)  033116  012746  007415                      MOV     #OPR004,-(SP)
 (7)  033122  012746  011045                      MOV     #FMTOP1,-(SP)
 (6)  033126  012746  000007                      MOV     #7,-(SP)
 (3)  033132  010600                              MOV     SP,R0
 (4)  033134  104417                              TRAP    C$PNTF
 (4)  033136  062706  000020                      ADD     #20,SP
3274  033142  012701  000024                      MOV     #20.,R1          ;INITIALIZE WAIT COUNT
3275  033146                              5$:     WAITMS  #50.             ;CALL WAIT
3276  033160  004737  016400                      JSR     PC,GSTATR        ;GET STATUS
3277  033164  033606                              60$
3278  033166  032737  020000  003056              BIT     #WLSTAT,T.MP     ;CHECK IF WRITE LOCK SET
3279  033174  001037                              BNE     7$               ;YES - SKIP
3280  033176                                      PRINTF  #FMT2,#BELL      ;RING BELL
```

```
  (8)  033176  012746  010721                    MOV     #BELL,-(SP)
  (7)  033202  012746  011153                    MOV     #FMT2,-(SP)
  (6)  033206  012746  000002                    MOV     #2,-(SP)
  (3)  033212  010600                            MOV     SP,R0
  (4)  033214  104417                            TRAP    C$PNTF
  (4)  033216  062706  000006                    ADD     #6,SP
3281   033222  005301                            DEC     R1              ;DEC COUNT
3282   033224  001350                            BNE     5$              ;SKIP IF NOT 0
3283   033226                                    PRINTF  #FMT23,#P2T17E,#BYPSNM,#OPR1A,<B,RLDRV+1> ;RPT BYPASSED
 (11)  033226  005046                            CLR     -(SP)
 (11)  033230  153716  003037                    BISB    RLDRV+1,(SP)
 (10)  033234  012746  007366                    MOV     #OPR1A,-(SP)
  (9)  033240  012746  007471                    MOV     #BYPSNM,-(SP)
  (8)  033244  012746  006777                    MOV     #P2T17E,-(SP)
  (7)  033250  012746  011762                    MOV     #FMT23,-(SP)
  (6)  033254  012746  000005                    MOV     #5,-(SP)
  (3)  033260  010600                            MOV     SP,R0
  (4)  033262  104417                            TRAP    C$PNTF
  (4)  033264  062706  000014                    ADD     #14,SP
3284   033270                                    EXIT    TST
  (3)  033270  104432                            TRAP    C$EXIT
  (3)  033272  000446                            .WORD   L10032-.
3285   033274  004537  023324          7$:       JSR     R5,DATGEN       ;GENERATE DATA
3286   033300  000001                            1                       ;PATTERN 1
3287   033302  012705  003040                    MOV     #L.CS,R5        ;GET ADDRESS OF L REGS
3288   033306  012715  000112                    MOV     #WTDATA,(R5)    ;LOAD WRITE COMMAND
3289   033312  053715  003036                    BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
3290   033316  042725  002000                    BIC     #BIT10,(R5)+    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
3291   033322  012725  004472                    MOV     #OBUFF,(R5)+    ;LOAD BUS ADDRESS
3292   033326  005025                            CLR     (R5)+           ;CYL 0, HD 0, SECTOR 0
3293   033330  012725  177600                    MOV     #177600,(R5)+   ;128 WORDS
3294   033334  012701  000454                    MOV     #300.,R1        ;SET WAIT COUNT FOR 30 MS
3295   033340  005037  003012                    CLR     DONE            ;CLEAR INTERRUPT FLAG
3296   033344  014562  000006                    MOV     -(R5),RLMP(R2)  ;LOAD RL REGS
3297   033350  014562  000004                    MOV     -(R5),RLDA(R2)
3298   033354  014562  000002                    MOV     -(R5),RLBA(R2)
3299   033360  014562  000000                    MOV     -(R5),RLCS(R2)
3300   033364                          10$:      WAITUS  #1
3301   033376  005737  003012                    TST     DONE            ;CHECK IF INTERRUPT
3302   033402  001013                            BNE     14$             ;YES - SKIP
3303   033404  005301                            DEC     R1              ;DEC WAIT COUNT
3304   033406  001366                            BNE     10$             ;LOOP IF NOT 0
3305   033410  004737  016224                    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
3306   033414  012603                            MOV     (SP)+,R3        ;GET RESULT MESSAGE
3307   033416                                    ERRHRD  1701.,,ERR1
  (4)  033416  104456                            TRAP    C$ERHRD
  (5)  033420  003245                            .WORD   1701
  (5)  033422  000000                            .WORD   0
  (5)  033424  012070                            .WORD   ERR1
3308   033426                                    EXIT    SUB
  (3)  033426  104432                            TRAP    C$EXIT
  (3)  033430  000164                            .WORD   L10033-.
3309   033432  004737  016430          14$:      JSR     PC,GSTAT        ;GET STATUS
3310   033436  033606                            60$
3311   033440  032737  040000  003050            BIT     #DRVERR,T.CS    ;TEST IF ANY ERROR SET
3312   033446  001006                            BNE     15$             ;YES - SKIP
```

```
 3313  033450  012703  010246                    MOV     #MDRERR,R3      ;SET RESULT MESSAGE POINTER
 3314  033454                                    ERRHRD  1702.,,ERR3     ;REPORT ERROR NOT SET
  (4)  033454  104456                            TRAP    C$ERHRD
  (5)  033456  003246                            .WORD   1702
  (5)  033460  000000                            .WORD   0
  (5)  033462  012204                            .WORD   ERR3
 3315  033464  032737  002000  003056  15$:     BIT     #WGESTAT,T.MP   ;TEST IF WGE SET
 3316  033472  001006                            BNE     18$             ;YES - SKIP
 3317  033474  012703  010325                    MOV     #MWGERR,R3      ;SET MESSAGE FOR WGE NOT SET
 3318  033500                                    ERRHRD  1704.,,ERR3
  (4)  033500  104456                            TRAP    C$ERHRD
  (5)  033502  003250                            .WORD   1704
  (5)  033504  000000                            .WORD   0
  (5)  033506  012204                            .WORD   ERR3
 3319  033510  042737  040000  003050  18$:     BIC     #DRVERR,T.CS    ;CLEAR DRIVE ERROR BIT
 3320  033516  042737  002000  003056           BIC     #WGESTAT,T.MP   ;CLEAR WGE BIT
 3321  033524  032737  157400  003056           BIT     #157400,T.MP    ;TEST IF ANY OTHER ERRORS
 3322  033532  001004                            BNE     16$             ;YES - GO REPORT
 3323  033534  032737  036000  003050           BIT     #36000,T.CS     ;TEST ANY ERRORS IN CS REG
 3324  033542  001405                            BEQ     17$             ;NO - SKIP
 3325  033544                          16$:      ERRHRD  1703.,,ERR6     ;REPORT ERRORS
  (4)  033544  104456                            TRAP    C$ERHRD
  (5)  033546  003247                            .WORD   1703
  (5)  033550  000000                            .WORD   0
  (5)  033552  012372                            .WORD   ERR6
 3326  033554  000414                            BR      60$             ;EXIT TEST
 3327  033556  004737  016400          17$:      JSR     PC,GSTATR       ;GET STATUS AND RESET ERROR
 3328  033562  033606                            60$
 3329  033564  004537  023324                    JSR     R5,DATGEN       ;GO GENERATE DATA
 3330  033570  000007                            7                       ;PATTERN 7
 3331  033572  004737  024014                    JSR     PC,XREAD        ;READ DATA
 3332  033576  033606                            60$
 3333  033600  004737  023464                    JSR     PC,DATCOM       ;COMPARE DATA
 3334  033604  033606                            60$
 3335  033606  012737  000002  003022  60$:      MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 3336  033614                          ENDSUB
  (3)  033614                          L10033:
  (3)  033614  104403                            TRAP    C$ESUB
 3337  033616  012737  000002  003024  T3204$:   MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 3338  033624                                    PRINTF  #FMTOP1,#OPR12,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQ RESET WRT L
 (13)  033624  005046                            CLR     -(SP)
 (13)  033626  153716  003037                    BISB    RLDRV+1,(SP)
 (12)  033632  012746  006142                    MOV     #DRVNAM,-(SP)
 (11)  033636  013746  003032                    MOV     RLBAS,-(SP)
 (10)  033642  012746  006131                    MOV     #BASADD,-(SP)
  (9)  033646  012746  007366                    MOV     #OPR1A,-(SP)
  (8)  033652  012746  007347                    MOV     #OPR12,-(SP)
  (7)  033656  012746  011045                    MOV     #FMTOP1,-(SP)
  (6)  033662  012746  000007                    MOV     #7,-(SP)
  (3)  033666  010600                            MOV     SP,R0
  (4)  033670  104417                            TRAP    C$PNTF
  (4)  033672  062706  000020                    ADD     #20,SP
 3339  033676  012701  001274                    MOV     #700.,R1        ;INITIALIZE WAIT COUNT
 3340  033702                          16$:      WAITMS  #1
 3341  033714  004737  016400                    JSR     PC,GSTATR       ;GET STATUS
 3342  033720  033616                            T3204$
```

```
 3343  033722  032737  020000  003056          BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK RESET
 3344  033730  001403                          BEQ     T3265$
 3345  033732  005301                          DEC     R1              ;DEC WAIT COUNT
 3346  033734  001362                          BNE     16$             ;LOOP IF NOT 0
 3347  033736  000727                          BR      T3204$          ;ELSE REPEAT MESSAGE
 3348  033740                          T3265$:
 3349  033740                          ENDTST
  (3)  033740                          L10032:
  (3)  033740  104401                          TRAP    C$ETST
 3350
```

```
                                        .SBTTL  *TEST 7          **ADJACENT CYLINDER INTERFERENCE
                                        BGNTST                   ;TEST 7
                                                                                              T7::
                        012737  007031  003016          MOV     #P2T18E,ERHEAD  ;SET ERROR HEADER
                        004737  021030                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
                        004737  016362                  JSR     PC,TSTINT       ;INITIALIZE TEST
                        004737  016400                  JSR     PC,GSTATR       ;CLEAR DRIVE
                                                        T3365$
                        005037  003236                  CLR     PASCNT          ;CLEAR PASS TO 0
                        012705  177776                  MOV     #-2,R5          ;SET R5
                        005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
                        001007                          BNE     1$              ;NO - SKIP
                        032737  000001  013722          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
                        001003                          BNE     1$              ;YES - SKIP
                        012705  177730                  MOV     #-40.,R5                ;ELSE SET R5 TO NEG 20
                        000402                          BR      9$              ;SKIP
                        012705  177770          1$:     MOV     #-10,R5         ;ELSE SET FOR NEG 4
                        012701  002510          9$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
                        012737  000010  002304          MOV     #10,JJJ         ;SET CLEAR COUNT
                        013721  013724          2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
                        005337  002304                  DEC     JJJ             ;DEC COUNT
                        001373                                  BNE     2$              ;LOOP UNTIL 0
                        004537  023324                  JSR     R5,DATGEN       ;GO GENERATE DATA
                        000011                          9.                      ;PATTERN 9
                        013737  013726  002512          MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
                        013737  013726  002514          MOV     HILIMW,T33TBL+4 ;INTO APPROPRIATE LOCATIONS
                        013737  013726  002520          MOV     HILIMW,T33TBL+10
                        013737  013726  002526          MOV     HILIMW,T33TBL+16
                        062705  000002          T3300$: ADD     #2,R5

                        032737  000001  013722          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
                        001034                          BNE     5$              ;YES - SKIP
                        005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
                        001403                          BEQ     3$              ;NO - SKIP
                        062705  000006                  ADD     #6,R5           ;ELSE BUMP CYLINDER POINTER BY 3
                        000402                          BR      6$              ;SKIP
                        062705  000044          3$:     ADD     #36.,R5         ;BUMP TO NEXT ENTRY
                        022737  000001  002302  6$:     CMP     #1,T.DRIVE
                        001404                          BEQ     44$
                        020537  000244                  CMP     R5,164.
                        103013                          BHIS    4$
                        000403                          BR      69$

                        020527  000122          44$:    CMP     R5,#82.
                        103007                          BHIS    4$

                        016537  002610  002304  69$:    MOV     CYLTBL(R5),JJJ
                        043737  002310  002304          BIC     CLRBYT,JJJ
                        001013                          BNE     8$
                        000137  032722          4$:     JMP     T3165$
                        005705                  5$:     TST     R5              ;TEST IF R5 0
                        001002                          BNE     7$              ;NO - SKIP
                        062705  000002                  ADD     #2,R5
                        023705  002306          7$:     CMP     HLMTW,R5        ;TEST IF ALL CYLINDERS USED
                        001767                          BEQ     4$              ;YES - EXIT TEST
                        010537  002304                  MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
```

```
3407  034236  023737  002304  013724  8$:     CMP     JJJ,LOLIMW      ;CHECK IF LOWER THAN LOLIMIT
3408  034244  103721                          BLO     T3300$          ;YES - SKIP
3409  034246  023737  002304  013726          CMP     JJJ,HILIMW      ;CHECK IF HIGHER THAN HILIMIT
3410  034254  101315                          BHI     T3300$          ;YES - SKIP
3411  034256  012703  002550                  MOV     #TBT,R3
3412  034262  013713  002304                  MOV     JJJ,(R3)
3413  034266  013763  002304  000006          MOV     JJJ,6(R3)
3414  034274  013763  002304  000010          MOV     JJJ,10(R3)
3415  034302  013763  002304  000012          MOV     JJJ,12(R3)
3416  034310  013763  002304  000016          MOV     JJJ,16(R3)
3417  034316  162737  000001  002304          SUB     #1,JJJ
3418  034324  013763  002304  000002          MOV     JJJ,2(R3)
3419  034332  013763  002304  000012          MOV     JJJ,12(R3)
3420  034340  062737  000002  002304          ADD     #2,JJJ
3421  034346  013763  002304  000004          MOV     JJJ,4(R3)
3422  034354  013763  002304  000014          MOV     JJJ,14(R3)
3423  034362  010337  003030                  MOV     R3,TBLSTR
3424  034366  004737  020720                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
3425  034372                          T3301$:
3426  034372                          BGNSUB
  (3) 034372                                                          T7.1:
  (3) 034372  104402                          TRAP    C$BSUB
3427  034374  042737  003760  003010          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
3428  034402  005737  003236                  TST     PASCNT          ;TEST IF PASS 0
3429  034406  001414                          BEQ     11$             ;YES - SKIP
3430  034410  023727  003236  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
3431  034416  001404                          BEQ     10$             ;YES - SKIP
3432  034420  002407                          BLT     11$             ;CHECK IF LESS THAN 4, IF YES CLEAR TO 0
3433  034422  012737  000004  003236          MOV     #4,PASCNT       ;ELSE SET TO 4
3434  034430  052737  000020  003010  10$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
3435  034436  000405                          BR      12$             ;SKIP
3436  034440  005037  003236          11$:    CLR     PASCNT          ;SET PASS COUNT TO 0
3437  034444  052737  000040  003010          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
3438  034452  012737  000003  003026  12$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
3439  034460  012701  002510                  MOV     #T33TBL,R1
3440  034464  012703  002550                  MOV     #TBT,R3
3441  034470  005037  003120          15$:    CLR     DESSEC          ;CLEAR TO SECTOR 0
3442  034474  012137  003106                  MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
3443  034500  004737  017326                  JSR     PC,XSEEK        ;DO SEEK
3444  034504  035062                          60$
3445  034506  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3446  034512  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3447  034516  035062                          60$
3448  034520  012337  003106                  MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
3449  034524  004737  017326                  JSR     PC,XSEEK        ;DO SEEK
3450  034530  035062                          60$
3451  034532  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3452  034536  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3453  034542  035062                          60$
3454  034544  004737  022634                  JSR     PC,VERPOS       ;VERIFY POSITION
3455  034550  035062                          60$
3456  034552  004737  024512          16$:    JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
3457  034556  034666                          32$                     ;"YES" RETURN
3458  034560  032737  000001  003026          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
3459  034566  001425                          BEQ     29$             ;NO - SKIP
3460  034570  004737  023754                  JSR     PC,XWRITE       ;DO WRITE
```

```
3461  034574  035062                          60$
3462  034576  005237  003120                  INC     DESSEC          ;INC SECTOR
3463  034602  022737  000050  003120          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3464  034610  001360                          BNE     16$             ;NO - SKIP
3465  034612  042737  000060  003010          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
3466  034620  042737  000001  003026          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
3467  034626  052737  000100  003010          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
3468  034634  005037  003120                  CLR     DESSEC          ;CLEAR TO SECTOR 0
3469  034640  000744                          BR      16$             ;SKIP
3470  034642  032737  000002  003026  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
3471  034650  001414                          BEQ     33$             ;NO - SKIP
3472  034652  004737  024014          31$:    JSR     PC,XREAD        ;ELSE DO READ
3473  034656  035062                          60$
3474  034660  004737  023464                  JSR     PC,DATCOM       ;COMPARE DATA
3475  034664  035062                          60$
3476  034666  005237  003120          32$:    INC     DESSEC          ;BUMP SECTOR
3477  034672  022737  000050  003120          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3478  034700  001324                          BNE     16$             ;NO - LOOP
3479  034702  005037  003120          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
3480  034706  005037  003026                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
3481  034712  005237  003236                  INC     PASCNT          ;BUMP PASS COUNT
3482  034716  042737  003760  003010          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
3483  034724  023727  003236  000004          CMP     PASCNT,#4       ;TEST IS PASS 4
3484  034732  001453                          BEQ     60$             ;YES - SKIP
3485  034734  023727  003236  000010          CMP     PASCNT,#8.      ;TEST IF PASS 8.
3486  034742  001447                          BEQ     60$             ;YES - SKIP
3487  034744  023727  003236  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3488  034752  001430                          BEQ     39$             ;YES - SKIP
3489  034754  023727  003236  000007          CMP     PASCNT,#7       ;TEST IF PASS 7
3490  034762  001430                          BEQ     40$             ;YES - SKIP
3491  034764  012737  000001  003026          MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED
3492  034772  023727  003236  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
3493  035000  001411                          BEQ     37$             ;YES - SKIP
3494  035002  023727  003236  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
3495  035010  001405                          BEQ     37$             ;YES - SKIP
3496  035012  052737  000040  003010          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
3497  035020  000137  034470          36$:    JMP     15$             ;GO DO NEXT PASS
3498  035024  052737  000020  003010  37$:    BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUALIFIER
3499  035032  000772                          BR      36$
3500  035034  052737  000200  003010  39$:    BIS     #REVSKS,OPFLAG  ;SET MESSAGE QUALIFIER
3501  035042  000403                          BR      41$
3502  035044  052737  000400  003010  40$:    BIS     #FWDSKS,OPFLAG  ;SET MESSAGE QUALIFIER
3503  035052  012737  000002  003026  41$:    MOV     #BIT1,WRTSWI    ;SET READ REQUIRED
3504  035060  000757                          BR      36$
3505  035062  012737  000002  003022  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3506  035070                        ENDSUB
 (3)  035070                        L10035:
 (3)  035070  104403                          TRAP    C$ESUB
3507  035072                                  ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  035072  104410                          TRAP    C$ESCAPE
 (3)  035074  000060                          .WORD   L10034-.
3508  035076  012737  000003  003026          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
3509  035104  023727  003236  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
3510  035112  001004                          BNE     45$             ;NO - SKIP
3511  035114  012737  002520  003030          MOV     #T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
3512  035122  000410                          BR      48$             ;GO START PASS 4
```

```
 3513  035124  005037  003236          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
 3514  035130  004737  020744                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 3515  035134  034110                          T3300$                  ;ABORT RETURN
 3516  035136  012737  002510  003030          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
 3517
 3518  035144  062703  000010          48$:    ADD     #10,R3
 3519  035150  000137  034372                  JMP     T3301$
 3520  035154                          T3365$:
 3521  035154                          ENDTST
  (3)  035154                          L10034:
  (3)  035154  104401                          TRAP    C$ETST
```

```
3523                                    .SBTTL  *TEST 8          **OVERWRITE
3524  035156                            BGNTST                   ;TEST 8
 (3)  035156                                                                  T8::
3525  035156  012737  007053  003016            MOV     #P2T19E,ERHEAD  ;SET ERROR HEADER
3526  035164  004737  021030                    JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
3527  035170  004737  016362                    JSR     PC,TSTINT       ;INITIALIZE TEST
3528  035174  004737  016400                    JSR     PC,GSTATR       ;CLEAR DRIVE
3529  035200  036346                            T3465$
3530  035202  005037  003236                    CLR     PASCNT          ;CLEAR PASS TO 0
3531  035206  012705  177776                    MOV     #-2,R5          ;SET R5
3532  035212  005737  003444                    TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
3533  035216  001007                            BNE     1$              ;NO - SKIP
3534  035220  032737  000001  013722            BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
3535  035226  001003                            BNE     9$              ;YES - SKIP
3536  035230  012705  177730                    MOV     #-40.,R5        ;ELSE SET R5 TO NEG 20
3537  035234  000402                            BR      9$              ;SKIP
3538  035236  012705  177770            1$:     MOV     #-10,R5         ;SET FOR NEXT ENTRY
3539  035242  012701  002510            9$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
3540  035246  012737  000010  002304            MOV     #10,JJJ         ;SET CLEAR COUNT
3541  035254  013721  013724            2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
3542  035260  005337  002304                    DEC     JJJ             ;DEC COUNT
3543  035264  001373                            BNE     2$              ;LOOP UNTIL 0
3544  035266  013737  013726  002512            MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
3545  035274  013737  013726  002516            MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
3546  035302  013737  013726  002522            MOV     HILIMW,T33TBL+12
3547  035310  062705  000002            T3400$: ADD     #2,R5
3548  035314  032737  000001  013722            BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
3549  035322  001034                            BNE     5$              ;YES - SKIP
3550  035324  005737  003444                    TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
3551  035330  001003                            BNE     3$              ;NO - SKIP
3552  035332  062705  000046                    ADD     #38.,R5         ;ELSE BUMP CYLINDER POINTER BY 19
3553  035336  000402                            BR      6$              ;SKIP
3554  035340  062705  000006            3$:     ADD     #6,R5           ;BUMP CYLINDER POINTER BY 3
3555  035344  022737  000001  002302    6$:     CMP     #1,T.DRIVE
3556  035352  001404                            BEQ     444$
3557  035354  020527  000244                    CMP     R5,#164.
3558  035360  103013                            BHIS    4$
3559  035362  000403                            BR      669$
3560  035364  020527  000122            444$:   CMP     R5,#82.
3561  035370  103007                            BHIS    4$
3562  035372  016537  002610  002304    669$:   MOV     CYLTBL(R5),JJJ
3563  035400  043737  002310  002304            BIC     CLRBYT,JJJ
3564  035406  001013                            BNE     8$
3565  035410  000137  036346            4$:     JMP     T3465$          ;EXIT TEST
3566  035414  005705                    5$:     TST     R5              ;TEST IF R5 0
3567  035416  001002                            BNE     7$              ;NO - SKIP
3568  035420  062705  000002                    ADD     #2,R5
3569  035424  022705  002306            7$:     CMP     #HLMTW,R5       ;TEST IF ALL CYLINDERS USED
3570  035430  001767                            BEQ     4$              ;YES - EXIT TEST
3571  035432  010537  002304                    MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
3572  035436  023737  002304  013724    8$:     CMP     JJJ,LOLIMW      ;TEST IF PAST LO LIMIT
3573  035444  103721                            BLO     T3400$          ;YES - SKIP
3574  035446  023737  002304  013726            CMP     JJJ,HILIMW      ;TEST IF PAST HILIMIT
3575  035454  101315                            BHI     T3400$          ;YES - SKIP
3576  035456  012703  002550                    MOV     #TBT,R3
3577  035462  013713  002304                    MOV     JJJ,(R3)
```

```
3578  035466  013763  002304  000002          MOV     JJJ,2(R3)
3579  035474  013763  002304  000004          MOV     JJJ,4(R3)
3580  035502  013763  002304  000006          MOV     JJJ,6(R3)
3581  035510  013763  002304  000010          MOV     JJJ,10(R3)
3582  035516  013763  002304  000012          MOV     JJJ,12(R3)
3583  035524  010337  003030                  MOV     R3,TBLSTR
3584  035530  004737  020720                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
3585  035534                          T3401$:
3586  035534                          BGNSUB
 (3)  035534                                                          T8.1:
 (3)  035534  104402                  TRAP    C$BSUB
3587  035536  042737  003760  003010          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
3588  035544  005737  003236                  TST     PASCNT          ;TEST IF PASS 0
3589  035550  001414                          BEQ     11$             ;YES - SKIP
3590  035552  023727  003236  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3591  035560  001404                          BEQ     10$             ;YES - SKIP
3592  035562  002407                          BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
3593  035564  012737  000003  003236          MOV     #3,PASCNT       ;ELSE SET TO 3
3594  035572  052737  000020  003010  10$:    BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
3595  035600  000405                          BR      12$             ;SKIP
3596  035602  005037  003236          11$:    CLR     PASCNT          ;SET PASS COUNT TO 0
3597  035606  052737  000040  003010          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
3598  035614  012737  000003  003026  12$:    MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
3599  035622  012701  002510                  MOV     #T3?TBL,R1
3600  035626  012703  002550                  MOV     #TBT,R3
3601  035632  005037  003120          15$:    CLR     DESSEC
3602  035636  012137  003106                  MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
3603  035642  004737  017326                  JSR     PC,XSEEK        ;DO SEEK
3604  035646  036254                          60$
3605  035650  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3606  035654  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3607  035660  036254                          60$
3608  035662  012337  003106                  MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
3609  035666  004737  017326                  JSR     PC,XSEEK        ;DO SEEK
3610  035672  036254                          60$
3611  035674  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3612  035700  004737  022222                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3613  035704  036254                          60$
3614  035706  004737  022634                  JSR     PC,VERPOS       ;VERIFY POSITION
3615  035712  036254                          60$
3616  035714  004737  024512          16$:    JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
3617  035720  036070                          32$                     ;"YES" RETURN
3618  035722  005737  003236                  TST     PASCNT          ;TEST IF PASS 0
3619  035726  001407                          BEQ     17$             ;YES - SKIP
3620  035730  022737  000003  003236          CMP     #3,PASCNT       ;TEST IF PASS 3
3621  035736  001403                          BEQ     17$             ;YES - SKIP
3622  035740  005037  035760                  CLR     25$             ;ELSE CLEAR DATA PATTERN SELECTOR
3623  035744  000403                          BR      18$
3624  035746  012737  000010  035760  17$:    MOV     #8.,25$         ;SET DATA PATTERN SELECTOR TO 8
3625  035754  004537  023324          18$:    JSR     R5,DATGEN       ;GO GENERATE DATA
3626  035760  000000                  25$:    .WORD   0
3627  035762  032737  000001  003026          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
3628  035770  001425                          BEQ     29$             ;NO - SKIP
3629  035772  004737  023754                  JSR     PC,XWRITE       ;DO WRITE
3630  035776  036254                          60$
3631  036000  005237  003120                  INC     DESSEC          ;INC SECTOR
```

```
3632  036004  022737  000050  003120          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3633  036012  001340                          BNE     16$             ;NO - SKIP
3634  036014  042737  000060  003010          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
3635  036022  042737  000001  003026          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
3636  036030  052737  000100  003010          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
3637  036036  005037  003120                  CLR     DESSEC          ;CLEAR TO SECTOR 0
3638  036042  000724                          BR      16$             ;SKIP
3639  036044  032737  000002  003026  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
3640  036052  001414                          BEQ     33$             ;NO - SKIP
3641  036054  004737  024014          31$:    JSR     PC,XREAD        ;ELSE DO READ
3642  036060  036254                          60$
3643  036062  004737  023464                  JSR     PC,DATCOM       ;COMPARE DATA
3644  036066  036254                          60$
3645  036070  005237  003120          32$:    INC     DESSEC          ;BUMP SECTOR
3646  036074  022737  000050  003120          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3647  036102  001304                          BNE     16$             ;NO - LOOP
3648  036104  005037  003120          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
3649  036110  005037  003026                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
3650  036114  005237  003236                  INC     PASCNT          ;BUMP PASS COUNT
3651  036120  042737  003760  003010          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
3652  036126  023727  003236  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
3653  036134  001447                          BEQ     60$             ;YES - SKIP
3654  036136  023727  003236  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
3655  036144  001443                          BEQ     60$             ;YES - SKIP
3656  036146  023727  003236  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
3657  036154  001424                          BEQ     39$             ;YES - SKIP
3658  036156  023727  003236  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
3659  036164  001424                          BEQ     40$             ;YES - SKIP
3660  036166  012737  000002  003026          MOV     #BIT1,WRTSWI    ;SET WRITE REQUIRED BIT
3661  036174  023727  003236  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
3662  036202  001405                          BEQ     37$             ;YES - SKIP
3663  036204  052737  001000  003010          BIS     #REVSKO,OPFLAG  ;SET REVERSE QUALIFIER
3664  036212  000137  035632          36$:    JMP     15$             ;GO DO NEXT PASS
3665  036216  052737  002000  003010  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
3666  036224  000772                          BR      36$             ;GO DO NEXT PASS
3667  036226  052737  000020  003010  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
3668  036234  000403                          BR      41$             ;SKIP
3669  036236  052737  000040  003010  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
3670  036244  012737  000001  003026  41$:    MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED BIT
3671  036252  000757                          BR      36$             ;GO DO NEXT PASS
3672  036254  012737  000002  003022  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3673  036262                          ENDSUB
 (3)  036262                          L10037:
 (3)  036262  104403                          TRAP    C$ESUB
3674  036264                          ESCAPE  TST                     ;EXIT TEST IF ERROR
 (3)  036264  104410                          TRAP    C$ESCAPE
 (3)  036266  000060                          .WORD   L10036-.
3675  036270  012737  000003  003026          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
3676  036276  023727  003236  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3677  036304  001004                          BNE     45$             ;NO - SKIP
3678  036306  012737  002516  003030          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
3679  036314  000410                          BR      48$             ;GO START PASS 4
3680  036316  005037  003236          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
3681  036322  004737  020744                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD ONE OR ABORT TEST
3682  036326  035310                          T3400$                  ;ABORT RETURN
3683  036330  012737  002510  003030          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
```

```
 3684  036336  062703  000006          48$:    ADD     #6,R3
 3685  036342  000137  035534                  JMP     T3401$
 3686  036346                          T3465$:
 3687  036346                          ENDTST
  (3)  036346                          L10036:
  (3)  036346  104401                         TRAP    C$ETST
 3688  036350                          ENDMOD
```

```
                                                .SBTTL  PARAMETER CODING
                                                BGNMOD  HRDPRM
                                                BGNHRD
                                                        .WORD L10040-L$HARD/2
                                                GPRML   CNTYPE,CNT,1,YES
                                                        .WORD   T$CODE
                                                        .WORD   CNTYPE
                                                        .WORD   1
                                                GPRMA   CSRMSG,CSR,0,160000,177776,YES
                                                        .WORD   T$CODE
                                                        .WORD   CSRMSG
                                                        .WORD   T$LOLIM
                                                        .WORD   T$HILIM
                                                GPRMA   VECMSG,VECT,0,0,776,YES
                                                        .WORD   T$CODE
                                                        .WORD   VECMSG
                                                        .WORD   T$LOLIM
                                                        .WORD   T$HILIM
                                                GPRMD   DRMSG,DRSB,0,3400,0,7,YES
                                                        .WORD   T$CODE
                                                        .WORD   DRMSG
                                                        .WORD   3400
                                                        .WORD   T$LOLIM
                                                        .WORD   T$HILIM
                                                GPRML   DRTYPE,TYPDR,1,YES
                                                        .WORD   T$CODE
                                                        .WORD   DRTYPE
                                                        .WORD   1
                                                GPRMD   BRMSG,PRIOR,0,340,0,7,YES
                                                        .WORD   T$CODE
                                                        .WORD   BRMSG
                                                        .WORD   340
                                                        .WORD   T$LOLIM
                                                        .WORD   T$HILIM

036432                                          ENDHRD
                                                        .EVEN
036432                                          L10040:

036432  052502  020123  042101                CSRMSG: .ASCIZ  /BUS ADDRESS/
036440  051104  051505  000123
036446  042526  052103  051117                VECMSG: .ASCIZ  /VECTOR/
036454     000
036455     102  020122  042514                BRMSG:  .ASCIZ  /BR LEVEL/
035462  042526  000114
036466  051104  053111  020105                DRTYPE: .ASCIZ  /DRIVE TYPE = RL01/
036474  054524  042520  036440
036502  051040  030114  000061
036510  051104  053111  000105                DRMSG:  .ASCIZ  /DRIVE/
036516  046122  030461     000                CNTYPE: .ASCIZ  /RL11/
036523                                          ENDMOD
        036524                                          .EVEN

036524                                          BGNMOD  SFTPRM
036524                                          BGNSFT
036524  000056                                          .WORD L10041-L$SOFT/2
```

```
3713
3715  036526                    GPRML   CYLQ,MISWI,1,YES
 (4)  036526  000130                    .WORD   T$CODE
 (4)  036530  036662                    .WORD   CYLQ
 (4)  036532  000001                    .WORD   1
3716  036534                    GPRML   SECQ,MISWI,2,YES
 (4)  036534  000130                    .WORD   T$CODE
 (4)  036536  036676                    .WORD   SECQ
 (4)  036540  000002                    .WORD   2
3722  036542                    GPRML   MANQ,MISWI,100000,YES
 (4)  036542  000130                    .WORD   T$CODE
 (4)  036544  036713                    .WORD   MANQ
 (4)  036546  100000                    .WORD   100000
3723
3725  036550                    GPRML   LOLIMQ,MISWI,40000,YES
 (4)  036550  000130                    .WORD   T$CODE
 (4)  036552  036747                    .WORD   LOLIMQ
 (4)  036554  040000                    .WORD   40000
3726  036556                    XFERF   1$
 (5)  036556  006044                    .WORD   T$CODE
3727  036560                    GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
 (4)  036560  001052                    .WORD   T$CODE
 (4)  036562  036766                    .WORD   LIMVAL
 (4)  036564  000377                    .WORD   255.
 (4)  036566  000000                    .WORD   T$LOLIM
 (4)  036570  000375                    .WORD   T$HILIM
3728  036572              1$:   GPRML   HILIMQ,MISWI,20000,YES
 (4)  036572  000130                    .WORD   T$CODE
 (4)  036574  036774                    .WORD   HILIMQ
 (4)  036576  020000                    .WORD   20000
3729  036600                    XFERF   2$
 (5)  036600  006044                    .WORD   T$CODE
3730  036602                    GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
 (4)  036602  002052                    .WORD   T$CODE
 (4)  036604  036766                    .WORD   LIMVAL
 (4)  036606  000377                    .WORD   255.
 (4)  036610  000000                    .WORD   T$LOLIM
 (4)  036612  000377                    .WORD   T$HILIM
3731  036614              2$:   GPRML   HEADQ,MISWI,10000,YES
 (4)  036614  000130                    .WORD   T$CODE
 (4)  036616  037015                    .WORD   HEADQ
 (4)  036620  010000                    .WORD   10000
3732  036622                    XFERF   3$
 (5)  036622  006044                    .WORD   T$CODE
3733  036624                    GPRMD   HEADV,HEAD,D,17,0,1,YES
 (4)  036624  003052                    .WORD   T$CODE
 (4)  036626  037037                    .WORD   HEADV
 (4)  036630  000017                    .WORD   17
 (4)  036632  000000                    .WORD   T$LOLIM
 (4)  036634  000001                    .WORD   T$HILIM
3735  036636              3$:   GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
 (4)  036636  004052                    .WORD   T$CODE
 (4)  036640  037062                    .WORD   ERLIMQ
 (4)  036642  000377                    .WORD   377
 (4)  036644  000000                    .WORD   T$LOLIM
 (4)  036646  000377                    .WORD   T$HILIM
```

```
3737  036650                                        GPRMD   DCLIMQ,DCLIM,D,377,1,377,YES
 (4)  036650  005052                                .WORD   T$CODE
 (4)  036652  037104                                .WORD   DCLIMQ
 (4)  036654  000377                                .WORD   377
 (4)  036656  000001                                .WORD   T$LOLIM
 (4)  036660  000377                                .WORD   T$HILIM
3739  036662                              ENDSFT
 (2)                                     .EVEN
 (3)  036662                              L10041:
3740
3742  036662  051525  020105  046101     CYLQ:   .ASCIZ  /USE ALL CYL/
       036670  020114  054503  000114
3743  036676  051525  020105  046101     SECQ:   .ASCIZ  /USE ALL SECT/
       036704  020114  042523  052103
       036712     000
3749  036713     104  020117  040515     MANQ:   .ASCIZ  /DO MANUAL INTERVENTION TEST/
       036720  052516  046101  044440
       036726  052116  051105  042526
       036734  052116  047511  020116
       036742  042524  052123     000
3751  036747     114  053517  051440     LOLIMQ: .ASCIZ  /LOW SEEK LIMIT/
       036754  042505  020113  044514
       036762  044515  000124
3752  036766  040526  052514  000105     LIMVAL: .ASCIZ  /VALUE/
3753  036774  050125  042520  020122     HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
       037002  042523  045505  046040
       037010  046511  052111     000
3754  037015     125  042523  047440     HEADQ:  .ASCIZ  /USE ONLY ONE SURF/
       037022  046116  020131  047117
       037030  020105  052523  043122
       037036     000
3755  037037     127  040510  020124     HEADV:  .ASCIZ  /WHAT SURF (0 OR 1)/
       037044  052523  043122  024040
       037052  020060  051117  030440
       037060  000051
3757  037062  047111  052520  020124     ERLIMQ: .ASCIZ  /INPUT ERROR LIMIT/
       037070  051105  047522  020122
       037076  044514  044515  000124
3759  037104  040504  040524  041440     DCLIMQ: .ASCIZ  /DATA CMP ERR LMT/
       037112  050115  042440  051122
       037120  046040  052115     000
3761          037126                           .EVEN
3762  037126                              ENDMOD
3763
3764  037126                              LASTAD
 (2)                                     .EVEN
 (4)  037126  000000                      .WORD   0
 (4)  037130  000000                      .WORD   0
 (3)  037132                              L$LAST::
3765
3766          000001                      .END
```

```
ADR    = 000020 G          92#
AFMID    003214            495#  2844  2857
AFMIDU   003216            496#  2845
ALLCYL= 000001             110#  3097  3110  3363  3381  3534  3548
ALLSEC= 000002             111#
ANYERR= 100000             163#  1625  1926  2190
ARMID    003220            497#  2846  2858
ARMIDU   003222            498#  2847
ASSEMB= 000010               8
BADADD= 004000             142#  2433  2469  2471
BAMSK  = 000060            174#
BANAM    006233            677#  2648
BASADD   006131            672#  1154  1337  1366  1379  1497  1978  2646  2851  3080  3273  3338
BELL     010721            815#  3280
BHSTAT= 000010             200#
BIT0   = 000001 G           92#  3182  3190  3458  3466  3491  3627  3635  3670
BIT00  = 000001 G           92#   110   132   151  2597
BIT01  = 000002 G           92#   111   131   152
BIT02  = 000004 G           92#   112   133   153
BIT03  = 000010 G           92#   113   134   154
BIT04  = 000020 G           92#   135
BIT05  = 000040 G           92#   136
BIT06  = 000100 G           92#   137
BIT07  = 000200 G           92#   138
BIT08  = 000400 G           92#   139
BIT09  = 001000 G           92#   140
BIT1   = 000002 G           92#  3194  3211  3470  3503  3639  3660
BIT10  = 002000 G           92#   141  1598  1717  2011  2285  2450  2707  3290
BIT11  = 004000 G           92#   142
BIT12  = 010000 G           92#   114   143
BIT13  = 020000 G           92#   115   144
BIT14  = 040000 G           92#   116   145
BIT15  = 100000 G           92#   117   146  1053
BIT2   = 000004 G           92#  2451
BIT3   = 000010 G           92#
BIT4   = 000020 G           92#
BIT5   = 000040 G           92#
BIT6   = 000100 G           92#  2540  2915
BIT7   = 000200 G           92#
BIT8   = 000400 G           92#  2913
BIT9   = 001000 G           92#  1017
BOE    = 000400 G           92#
BRMSG    036455            3698  3704#
BSCHK    024512            2523# 2995  3176  3456  3616
BSFLAG   003024             427# 2526* 2546* 2553
BSFVAL   003500             530# 1975  2949*
BSNSTR   007540             772# 1977
BYPSNM   007471             769# 3283
CAFDT    011026             833# 2024  2493
CAMSK    002316             255# 1291* 1299*
CCYLUP   011017             832#
CHOSHD   020720            1945# 2698  2963  3021  3142  3424  3584
CKBSVD   021030            1975# 2959  3089  3355  3526
CKDATA= 000102              120#
CKERLM   016032             933   947   961   976   991  1096  1110  1132  1146  1160  1492#
CLKADR   003476             526# 1220*
```

```
CLKCSB= 172542            215#  2725*  3051*
CLKCSR= 172540            214#  2725*  2730*  3051*  3058*
CLKCTR= 172544            216#  2725*  2730   3051*  3058
CLKFLG  003474            525#  1219*  1222*  2676   3008
CLNCOD  015272 G         1390#
CLRBYT  002310            252#  1294*  1303*  3124   3398   3563
CLRPAR  026112           2614   2654#
CNT   = 000012             99#  3693
CNTYPE  036516           3693   3707#
COMPOP= 007777            130#  2593
CONHNG= 000004            153#
CONTIN  014164           1257#
COSTAT= 000040            202#
COUNT   003240            509#  2704*  2742*
CRDYMS= 000200            172#  1015   1327   1373   1396   1513   1524   2293   2299
CSNAM   006226            676#  2648
CSR   = 000000             94#  3694
CSRMSG  036432           3694   3702#
CURCYL  003110            459#  1679   1696   1840   1844   1845   1863   1869   1877   2226*  2251   2429   2456
                         2613   2650
CYLQ    036672           3715   3742#
CYLTBL  002610            334#  3123   3397   3562
CYLUP = 000004            133#
CYLWD   007456            767#  2613   2648
C$AU  = 000052              8#
C$AUTO= 000061              8#  1384
C$BRK = 000022              8#
C$BSEG= 000004              8#  1835
C$BSUB= 000002              8#  2976   3145   3248   3426   3586
C$CEFG= 000045              8#
C$CLCK= 000062              8#  1220
C$CLEA= 000012              8#  1409
C$CLOS= 000035              8#
C$CLP1= 000006              8#
C$CVEC= 000036              8#  1383   1402   1406
C$DCLN= 000044              8#  1340   1500
C$DODU= 000051              8#  1339   1370   1382   1499
C$DRPT= 000024              8#
C$DU  = 000053              8#  1413
C$EDIT= 000003              8#    63
C$ERDF= 000055              8#
C$ERHR= 000056              8#  1583   1620   1633   1639   1744   1749   1858   1874   1917   1928   2025   2036
                         2040   2048   2097   2109   2116   2178   2188   2192   2253   2306   2311   2415   2473
                         2488   2494   2503   2507   2718   2722   2936   2943   2946   3043   3047   3063   3067
                         3307   3314   3318   3325
C$ERRO= 0000 0              8#
C$ERSF= 000054              8#
C$ERSO= 000057              8#
C$ESCA= 000010              8#  2987   3225   3507   3674
C$FSEG= 000005              8#  1882
C$ESUB= 000003              8#  2986   3224   3336   3506   3673
C$ETST= 000001              8#  2860   2954   3001   3083   3238   3349   3521   3687
C$EXIT= 000032              8#  2873   3012   3284   3308
C$GETB= 000026              8#
C$GETW= 000027              8#
C$GMAN= 000043              8#
```

```
C$GPHR= 000042          8#  1275
C$GPLO= 000030          8#
C$GPRI= 000040          8#
C$INIT= 000011          8#  1343
C$INLP= 000020          8#  1494  1836  2930
C$MANI= 000050          8#  1225
C$MEM = 000031          8#
C$MSG = 000023          8#   934   948   962   977   992  1097  1111  1133  1147  1161
C$OPEN= 000034          8#
C$PNTB= 000014          8#  1035  1068  1082  1091  1154  1155  1157  2400  2568  2569  2573  2586
                      2602  2606  2610  2613  2625  2634  2635  2638  2646  2648  2649  2650
C$PNTF= 000017          8#  1336  1337  1338  1365  1366  1368  1377  1379  1381  1496  1497  1498
                      1977  1978  1979  2678  2850  2851  2852  2853  2854  2855  2856  2857  2858
                      3010  3079  3080  3081  3273  3280  3283  3338
C$PNTS= 000016          8#
C$PNTX= 000015          8#
C$QIO = 000377          8#
C$RDBU= 000007          8#  1427  1447
C$REFG= 000047          8#  1230  1234  1255  1258  1261
C$RESE= 000033          8#  1224  1407
C$REVI= 000003          8#    63
C$RFLA= 000021          8#
C$RPT = 000025          8#
C$SEFG= 000046          8#
C$SPRI= 000041          8#  1223  1309  1395
C$SVEC= 000037          8#  1308  1359  1393
C$TPRI= 000013          8#
C10MS   011000        830#
C5SEC   011037        834#  2035  2187  2502
C500MS  011011        831#  1916
DANAM   006240        678#  2648
DATACM= 000001        132#  2385  2396
DATCOM  023464       2376#  2983  3198  3271  3333  3474  3643
DATGEN  023324       2336#  2977  3018  3180  3265  3285  3329  3373  3625
DCKERR= 004000        168#  1045  1069
DCLIM = 000012        107#  3737
DCLIMQ  037104       3737   3759#
DCLIMW  013734       1198#  2412
DESDIF  003112        460#  1689* 1703* 1712* 1719  2610
DESHD   003116        462#  1727  1945* 1948* 1955  1957* 2102  2288  2431  2461  2538  2610  2613
                     2650   2868* 3253*
DESSEC  003120        463#  2464  2541  2613  2882* 2926* 2932  2940* 2964* 3020* 3161* 3178  3186*
                     3187   3192* 3200* 3201  3203* 3254* 3441* 3462* 3463  3468* 3476* 3477  3479*
                     3601*  3631* 3632  3637* 3645* 3646  3648*
DESSGN  003114        461#  1688* 1699* 1702* 1708* 1724  1867  2610  2735
DIAGMC= 000000           8
DIFAUG  003102        456#  1675* 1683* 1694* 1704  1712
DIFWD   007432        763#  2610
DIRBIT= 000004        183#  1726
DIRMSK  002320        256#  1292* 1300*
DLTERR= 010000        166#  1049
DONE    003012        422#  1480* 1516  1529  1596* 1603  1733* 1740  2008* 2019  2476* 2484  2714
                     2724*  3039  3050* 3059  3295* 3301
DRDYMS= 000001        175#  1330  1375  1399  1567  1612  1910  1921  2021  2029  2171  2182  2490
                     2498
DRMSG   036510       3696   3706#
```

```
DRSB   = 000010            98#  3696
DRSELT= 000004            112#  1227
DRSET  = 000010            188#  1543  1559  1607  2589
DRTYPE  036466            3697  3705#
DRVCNT  003100             455#  1240*  1264  1268*  1271*
DRVERR= 040000             164#  1561  1577  1631  3311  3319
DRVNAM  006142             673#  1154  1337  1366  1379  1497  1978  2646  2851  3080  3273  3338
DRVNAV  006147             674#
DSESTA= 000400             204#
DSMSK  = 001400             171#
DSPCOD  013736 G           1202#
EF.CON= 000036 G            92#  1258
EF.NEW= 000035 G            92#  1261
EF.PWR= 000034 G            92#  1230
EF.RES= 000037 G            92#  1255
EF.STA= 000040 G            92#  1234
ERHEAD  003016              424#  2573  2674*  2864*  2958*  3006*  3088*  3249*  3354*  3525*
ERLIM  = 000010             106#  3735
ERLIMQ  037062             3735  3757#
ERLIMW  013732             1197#  1492  1496
ERRCNT  003244              511#  1242  1247  1267  1929*  2193*  2952*
ERRPOI  003242              510#   925*   937*   951*   965*   980*   997*  1100*  1114*  1136*  1152*  1247*  1267*
                           1272*  1492
ERRSWI  003022              426#  1558*  1640*  1653  1655  1674*  1745*  1750*  1756  1758  1831*  1859*  1875*
                           1887  1889  1906*  1930*  1936  1938  1998*  2031*  2049*  2062  2064  2082*  2098*
                           2110*  2117*  2136  2138  2168*  2194*  2200  2202  2231  2233  2248*  2254*  2258
                           2260  2277*  2307*  2312*  2324  2326  2397  2406  2408  2416*  2504*  2508*  2514
                           2516  2891*  2948*  2985*  3082*  3223*  3335*  3337*  3505*  3672*
ERRVEC  003234              505#  1359  1383  1393  1406
ERR1    012070 G            922#  1583  1620  1639  1744  1874  2048  2306  2473  2488  2718  2936  2943
                           2946  3043  3063  3307
ERR10   013464 G           1148#  2097  2109  2116  2415
ERR2    012136 G            936#
ERR3    012204 G            950#  2178  3314  3318
ERR4    012252 G            964#
ERR5    012322 G            979#  1917  2025  2036  2188  2494  2503
ERR6    012372 G            994#  1633  1749  1858  1928  2040  2192  2311  2507  2722  3047  3067  3325
ERR7    013254 G           1099#
ERR8    013324 G           1113#  2253
ERR9    013420 G           1135#
EVL    = 000004 G            92#
EXACYL  003230              502#  2857  2858
EXHCYL  003226              501#  2855  2856
EXOCYL  003224              500#  2853  2854
EXROT   003232              503#  3081
E$END  = 002100               8#
E$LOAD= 000035                8#    63
FBSFIL  003676              533#  1981*  2527  2548  2883
FMTOP1  011045              836#  3273  3338
FMTOP2  011074              837#
FMTOP3  011116              838#
FMT1    011137              839#  2586  2638
FMT1.1  011144              840#  2625  2850  3079
FMT11   011363              849#  2628
FMT12   011371              850#  2631
FMT13   011377              851#  2610
```

```
FMT14   011443              852#  1155
FMT15   011475              853#  1082  1157
FMT16   011531              854#  2569
FMT17   011542              855#  1068
FMT18   011564              856#  2852
FMT19   011616              857#  2853  2854
FMT2    011153              841#  2602  2606  3280
FMT20   011653              858#  2855  2856
FMT21   011703              859#  2857  2858
FMT22   011726              860#  2613
FMT23   011762              861#  3283
FMT24   011776              862#  1336  1365  1377
FMT25   012003              863#  1496
FMT26   012013              864#  3081
FMT27   012037              865#  1091  2400
FMT28   012056              866#  1035
FMT3    011156              842#  1338  1368  1381  1498  1979
FMT4    011161              843#  2573
FMT5    011172              844#  1154  1337  1366  1379  1497  1978  2646  2851  3080
FMT6    011212              845#  2648
FMT7    011254              846#  2650
FMT8    011324              847#  2649
FMT9    011356              848#  1977  2568  2678  3010
FOLWRT= 000100              137#   147  3191  3467  3636
FRMWD   007463              768#  2610
FWDSKO= 002000              141#   147  3217  3665
FWDSKS= 000400              139#   147  3502
F$AU  = 000015                8#
F$AUTO= 000020                8#  1357  1384
F$BGN = 000040                8#    58    65    90   218   224   634   645   868   877   922   936   950
                            964   979   994  1099  1113  1135  1148  1162  1165  1173  1182  1184  1200
                           1202  1209  1215  1216  1344  1357  1390  1391  1411  1415  1420  1462  1470
                           1835  2664  2669  2673  2860  2863  2873  2954  2957  2976  2986  2987  3001
                           3005  3012  3083  3087  3145  3224  3225  3238  3241  3248  3284  3308  3336
                           3349  3353  3426  3506  3507  3521  3524  3586  3673  3674  3687  3688  3691
                           3692  3708  3711  3712  3762
F$CLEA= 000007                8#  1391  1409
F$DU  = 000016                8#  1411  1413
F$END = 000041                8#    58    65    90   218   224   634   645   868   877   934   948   962
                            977   992  1097  1111  1133  1147  1161  1162  1173  1182  1184  1200  1202
                           1209  1215  1343  1344  1384  1390  1409  1413  1415  1420  1468  1484  1882
                           2664  2669  2673  2860  2863  2873  2954  2957  2976  2986  2987  3001  3005
                           3012  3083  3087  3145  3224  3225  3238  3241  3248  3284  3308  3336  3349
                           3353  3426  3506  3507  3521  3524  3586  3673  3674  3687  3688  3691  3700
                           3708  3711  3739  3762
F$HARD= 000004                8#  3692  3700  3726  3729  3732
F$HW  = 000013                8#  1174  1181
F$INIT= 000006                8#  1216  1343
F$JMP = 000050                8#  2873  3012  3284  3308
F$MOD = 000000                8#    58    65    90   218   224   634   645   868   877  1162  1173  1182
                           1184  1200  1202  1209  1215  1344  1390  1415  1420  2664  2669  3688  3691
                           3708  3711  3762
F$MSG = 000011                8#   922   934   936   948   950   962   964   977   979   992   994  1097
                           1099  1111  1113  1133  1135  1147  1148  1161
F$PROT= 000021                8#  1165  1169
F$PWR = 000017                8#
```

```
F$RPT = 000012              8#
F$SEG = 000003              8#  1835    1882
F$SOFT= 000005              8#  3712    3726    3729    3732    3739
F$SRV = 000010              8#  1462    1468    1470    1484
F$SUB = 000002              8#  2976    2986    3145    3224    3248    3336    3426    3506    3586    3673
F$SW  = 000014              8#  1185    1199
F$TEST= 000001              8#  2673    2860    2863    2954    2957    3001    3005    3083    3087    3238    3241    3349
                          3353  3521    3524    3687
GBND    002314            254#  1290*   1298*
GETPOS  022506           1677   1832    1838    1861    2210#   2249
GETSTA= 000003            187#  1010    1543    1546    1592
GLBDAT  002230 G          224#
GLBEQA  002230 G           90#
GLBERR  012070 G          877#
GLBSUB  015424 G         1420#
GLBTXT  005350 G          645#
GSTAT   016430           1548#  1565    1617    1623    1908    1919    2027    2169    2180    2496    3309
GSTATC  016414           1545#
GSTATG  016440           1544   1547    1550#
GSTATR  016400           1542#  2682    2866    2961    3016    3091    3251    3276    3327    3341    3357    3528
GTSTAT= 000104            121#  1599
G$CNTO= 000200              8#
G$DELM= 000372              8#  1429    1434    1450    1454
G$DISP= 000003              8#
G$EXCP= 000400              8#
G$HILI= 000002              8#
G$LOLI= 000001              8#
G$NO  = 000000              8#
G$OFFS= 000400              8#  3693    3694    3695    3696    3697    3698    3715    3716    3722    3725    3727    3728
                          3730  3731    3733    3735    3737
G$OFSI= 000376              8#  3693    3694    3695    3696    3697    3698    3715    3716    3722    3725    3727    3728
                          3730  3731    3733    3735    3737
G$PRMA= 000001              8#  3694    3695
G$PRMD= 000002              8#  3696    3698    3727    3730    3733    3735    3737
G$PRML= 000000              8#  3693    3697    3715    3716    3722    3725    3728    3731
G$RADA= 000140              8#
G$RADB= 000000              8#
G$RADD= 000040              8#  3727    3730    3733    3735    3737
G$RADL= 000120              8#  3693    3697    3715    3716    3722    3725    3728    3731
G$RADO= 000020              8#  3694    3695    3696    3698
G$XFER= 000004              8#  3726    3729    3732
G$YES = 000010              8#  3693    3694    3695    3696    3697    3698    3715    3716    3722    3725    3727    3728
                          3730  3731    3733    3735    3737
HADONE  003014            423#  1249*
HCESTA= 040000            210#  1571
HCRCER= 004000            169#  1032    1039
HDALIG= 000010            113#  1227
HDCYL   002322            257#  1293*   1301*
HDHSEL= 000100            196#  2101    2104
HDMOVF  007313            740#  1873
HDRCMP= 000002            131#  2083
HDR40 = 100000            146#  2279    2603
HDSEC = 000077            195#  2123    2126
HDSEL = 000020            184#  1729    2290
HDWD    007445            765#  2610    2613    2648
HDWRD1  003056            443#  1120    2145    2220
```

```
HDWRD2  003060            444#  2042
HDWRD3  003062            445#
HEAD  = 000006            105#  3733
HEADLM= 010000            114#  1946  1953  2869
HEADQ   037015           3731   3754#
HEADV   037037           3733   3755#
HEADW   013730           1196#  1948  2871
HFIN    003174            487#  2828  2855
HFINU   003176            488#  2829
HFOUT   003200            489#  2806  2855
HFOUTU  003202            490#  2807  2809
HICYL = 020000            115#  1305
HILIM = 000004            104#  3730
HILIMQ  036774           3728   3753#
HILIMW  013726           1195#  1307*  3106  3107  3108  3127  3132  3375  3376  3377  3378  3409  3544
                         3545   3546   3574
HLMTW   002306            251#  1289*  1297*  1307  1680  1682  1684  1706  2429  2794  2832  2848  2874
                         3404   3569
HNFERR= 010000            167#  1030
HOE   = 100000 G           92#
HOSTAT= 000020            201#  1569
HPTCOD  01370< G         1173#
HRDPRM  036350 G         3691#
HRDWTS  026142 G         2669#
HRIN    003204            491#  2830  2856
HRINU   003206            492#  2831
HROUT   003210            493#  2808  2856
HROUTU  003212            494#
HSMSK = 000100            179#  2463
HSSTAT= 000100            203#  2149
IBE   = 010000 G           92#
IBUFF   004072            535#  1077  2085  2280  2388  2453  2890  2918
IDU   = 000040 G           92#
IER   = 020000 G           92#
INITCO  013760 G         1215#
INOUTS  000020            135#   147  3153  3189  3219  3434  3465  3498  3594  3634  3667
INTBL= 000100             173#
INTHLR  015752           1308   1474#
ISR   = 000100 G           92#
IXE   = 004000 G           92#
I$AU  = 000041              8#
I$AUTO= 000041              8#  1357#  1384#
I$CLN = 000041              8#  1391#  1409#
I$DU  = 000041              8#  1411#  1413#
I$HRD = 000041           3692#  3700#
I$INIT= 000041              8#  1216#  1343#
I$MOD = 000041              8#    58#    65#   90#  218#  224#  634#  645#  868#  877#  1162#  1173#  1182#
                         1184#  1200#  1202#  1209#  1215#  1344#  1390#  1415#  1420#  2664#  2669#  3688#  3691#
                         3708#  3711#  3762#
I$MSG = 000041              8#   922#   934#   936#  948#  950#  962#  964#  977#  979#  992#  994#  1097#
                         1099#  1111#  1113#  1133#  1135#  1147#  1148#  1161#
I$PROT- 000040              8#  1165#
I$PTAB= 000041              8#
I$PWR = 000041              8#
I$RPT = 000041              8#
I$SEG = 000041              8#  1835#  1882#  2673  2863  2957  2976  3005  3087  3145  3241  3248  3353
```

```
                    3426    3524    3586
I$SFTU= 000041       8#
I$SFT = 000041      3712#   3739#
I$SRV = 000041       8#     1462#   1468#   1470#   1484#
I$SUB = 000041       8#     2673    2863    2957    2976#   2986#   3005    3087    3145#   3224#   3241    3248#   3308
                    3336#   3353    3426#   3506#   3524    3586#   3673#
I$TST = 000041       8#     2673#   2860#   2863#   2873    2954#   2957#   2976    2987    3001#   3005#   3012    3083#
                    3087#   3145    3225    3238#   3241#   3248    3284    3349#   3353#   3426    3507    3521#   3524#
                    3586    3674    3687#
JJJ     002304       250#   3102*   3104*   3123*   3124*   3129*   3130    3132    3135    3136    3137    3138    3139
                    3140    3369*   3371*   3397*   3398*   3406*   3407    3409    3412    3413    3414    3415    3416
                    3417*   3418    3419    3420*   3421    3422    3540*   3542*   3562*   3563*   3571*   3572    3574
                    3577    3578    3579    3580    3581    3582
J$JMP = 000167       8#
LAB     014136      1250#
LABACF  007263       737#   2857
LABACR  007277       738#   2858
LABEXP  007176       732#   2852
LABHCF  007233       735#   2855
LABHCR  007247       736#   2856
LABIN   007153       729#   2852
LABMID  007161       730#   2852
LABOCF  007207       733#   2853
LABOCR  007221       734#   2854
LABOUT  007170       731#   2852
LAB1    006252       680#   2649
LAB2    006265       681#   2650
LIMVAL  036766      3727    3730    3752#
LOCERR  003450       514#   2886*   2928*   2935*   2950
LOCYL = 040000       116#   1250
LOE   = 040000 G      92#
LOLIM = 000002       103#   3727
LOLIMQ  036747      3725    3751#
LOLIMW  013724      1194#   1252*   3022    3103    3130    3370    3407    3541    3572
LOT   = 000010 G      92#
L$ACP   002110 G      63#
L$APT   002036 G      63#
L$AUT   002070 G      63#
L$AUTO  014734 G      63     1357#
L$CCP   002106 G      63#
L$CLEA  015272 G      63     1391#
L$CO    002032 G      63#
L$DEPO  002011 G      63#
L$DESC  002122 G      63      66#
L$DESP  002076 G      63#
L$DEVP  002060 G      63#
L$DISP  013740 G      63     1207#
L$DLY   002116 G      63#    1423*   1429    1433*   1434    1441*   1449*   1450    1454
L$DTP.  002040 G      63#
L$DTYP  002034 G      63#
L$DU    015420 G      63     1411#
L$DUT   002072 G      63#
L$DVTY  002216 G      63      67#
L$EF    002052 G      63#
L$ENVI  002044 G      63#
L$ETP   002102 G      63#
```

```
L$EXP1  002046 G          63#
L$EXP4  002064 G          63#
L$EXP5  002066 G          63#
L$HARD  036352 G          63    3692#
L$HIME  002120 G          63#
L$HPCP  002016 G          63#
L$HPTP  002022 G          63#
L$HW    013704 G          63    1174#
L$ICP   002104 G          63#
L$INIT  013760 G          63    1216#
L$LADP  002026 G          63#
L$LAST  037132 G          63    3704#
L$LOAD  002100 G          63#
L$LUN   002074 G          63#
L$MREV  002050 G          63#
L$NAME  002000 G          63#
L$PRIO  002042 G          63#
L$PROT  013674 G          63    1165#
L$PRT   002112 G          63#
L$REPP  002062 G          63#
L$REV   002010 G          63#
L$SOFT  036526 G          63    3712#
L$SPC   002056 G          63#
L$SPCP  002020 G          63#
L$SPTP  002024 G          63#
L$STA   002030 G          63#
L$SW    013722 G          63    1185#
L$TEST  002114 G          63#
L$TIML  002014 G          63#
L$UNIT  002012 G          63#   1232    1240    1268
L.BA    003042          436#   2286*   2292    2649
L.CS    003040          435#   1597*   1598*   1599*   1601    1643    1714    2009    2052    2283*   2284*   2285*   2297
                        2447   2575    2649    2711    2732*   3287
L.DA    003044          437#   1592*   1594*   1600    1607    2287*   2290*   2291    2589    2649    2710
L.MP    003046          438#   1588    2001    2649    3031    3049
L10000  012134          934#
L10001  012202          948#
L10002  012250          962#
L10003  012320          977#
L10004  012370          992#
L10005  013252         1097#
L10006  013322         1111#
L10007  013416         1133#
L10010  013462         1147#
L10011  013672         1161#
L10013  013720         1174   1181#
L10014  013736         1185   1199#
L10015  014732         1343#
L10016  015270         1384#
L10017  015416         1409#
L10020  015422         1413#
L10021  015750         1468#
L10022  016030         1484#
L10023  030052         2860#
L10024  030570         2873   2954#
L10025  031004         2987   3001#
```

```
L10026  030730            2986#
L10027  031612            3012    3083#
L10030  032722            3225    3238#
L10031  032636            3224#
L10032  033740            3284    3349#
L10033  033614            3308    3336#
L10034  035154            3507    3521#
L10035  035070            3506#
L10036  036346            3674    3687#
L10037  036262            3673#
L10040  036432            3692    3700#
L10041  036662            3712    3739#
MAJINC  003472             524#   1443*   1457
MANQ    036713            3722    3749#
MAPROX  007143             728#   3081
MBADAD  006012             247     668#
MBADSF  006033             669#   2934
MBSETO= 000001             182#   1730
MCERR   010135             260     782#
MCONHN  006377             686#   1527
MCYLOC  010511             808#   1119    2629
MCYLUP  005550             238     660#
MDATCP  005432             236     653#
MDCRC   010157             784#   1047
MDHEDR  002000 G            58#
MDLT    010204             786#   1051
MDRDY   010124             781#   1915    2023    2177    2492
MDRERR  010246             261     788#   3313
MDRRES  006326             684#   1521
MDRVST  010274             798#   1104
MDSERR  010257             275     797#
MERRS   010714             814#   1035
MEXERS  010657             813#   1496
MFLERR  010436             263     806#
MFMTER  006063             670#   2942
MFOLWR  005620             242     663#
MFWDSK  005671             244     665#
MFWSKO  005722             246     666#
MGTSTA  005420             230     652#
MHCERR  010356             269     802#
MHCRC   010147             783#   1037
MHDERR  010421             264     805#
MHDRCP  005451             237     654#
MHFCRC  010216             787#   1043
MHNF    010170             785#   1041
MININC  003462             520#   1425*   1437
MINOUT  005577             240     662#
MISWI = 000000             102#   3715    3716    3722    3725    3728    3731
MISWIW  013722            1186#   1227*   1250    1305    1946    1953    2869    3097    3110    3244    3363    3381    3534
                          3548
MITEST= 100000             117#   1227
MNDRST  010516             266     809#   2626
MNEERR  010464             262     807#
MNOCLR  006413             687#
MNOINT  005344             685#   1532
MOPER   005517             657#   2586
```

```
MOPERR  010411   265    804#  1034
MORECE  003020   425#   1073*  1080   1083*  1088   1091   1150   1158*  1538*  2084*  2386*  2400   2412
                2419*
MOUTIN  005560   241    661#
MPNAM   006245   679#   2648
MQUALS= 003760   147#   3146   3206   3427   3482   3587   3651
MREAD   005354   234    648#
MREADH  005365   232    649#
MRESKO  005756   245    667#
MREVSK  005640   243    664#
MRLFAL  010606   811#   1619
MRSLT   005526   658#   1035   1155   2625
MSEEK   005350   231    647#
MSPERR  010307   272    799#
MSTERR  010342   271    801#
MTMBS   006110   671#   2945
MTOSLO  006306   683#   1518
MULOAD  005537   239    659#
MUNDEF  010541   810#   1582
MWDERR  010374   268    803#
MWGERR  010325   273    800#   3317
MWORD   006300   682#   1082   1155   1157
MWRCHK  005375   229    650#
MWRITE  005406   233    651#
MWRSET  005503   235    656#
MWRTAB  010645   812#   2472
M40HDR  005467   248    655#
NEWCYL  003106   458#   1127   1680   1682*  1683   1684*  1687*  1691   1693*  1694   1695*  1697   1706
                1840   1848*  1850*  1863   1869   1877   1966   1967*  2095   2251   2533   2690*  2703*
                2760*  2763*  2775*  2778*  2782*  2794*  2798*  2810*  2813*  2817*  2820*  2832*  2836*
                2848*  2874*  2965*  2998*  3022*  3162*  3168*  3255*  3442*  3448*  3602*  3608*
NOCLR = 000010   154#
NOCTLR  007635   774#   1365
NOERCT  003451   515#    923    995   1537*  2885*
NOIRPT= 000002   152#
NOOP  = 000100   127#
NOPWR   006166   675#   1336
NOTRDY  007673   775#   1377
NOTST1  007750   776#   2678
NOTST4  010036   777#   3010
NXMERR= 020000   165#
NXTHL   002312   253#   1288*  1302*  2782
NXTPAS  014204  1263#   1278   1280
OBUFF   004472   536#   1076   2339   2355   2357   2364   2387   2455   3291
OFIN    003144   475#   2684   2790   2853
OFINU   003146   476#   2791
OFMID   003150   477#   2771   2853
OFMIDU  003152   478#   2772
OFOUT   003154   479#   2699   2853
OFOUTU  003156   480#   2700
OLDCYL  003104   457#   1679*  1844*  1965   1966*  2610
ONSWAP  021004  1842   1964#  2744
OPFLAG  003010   421#   1536*  1575*  1609   2083*  __79*  2385*  2396*  2433*  2469   2471*  2474*  2574*
                2579*  2582*  2585*  2593   2595   2603   2608   2611   3146*  3153*  3156*  3189*  3191*
                3206*  3217*  3219*  3221*  3427*  3434*  3437*  3465*  3467*  3482*  3496*  3498*  3500*
                3502*  3587*  3594*  3597*  3634*  3636*  3651*  3663*  3665*  3667*  3669*
```

```
OPIFRR= 002000            170#   1028   1071   1519
OPMSGS  002230            228#   2586   2602   2606
OPR004  007415            762#   3273
OPR1A   007366            759#   3273   3283   3338
OPR1B   007372            760#
OPR12   007347            758#   3338
ORIN    003160            481#   2792   2854
ORINU   003162            482#   2793
ORMID   003164            483#   2773   2854
ORMIDU  003166            484#   2774
OROUT   003170            485#   2701   2854
OROUTU  003172            486#   2702
OUTINS= 000040            136#    147   3156   3189   3221   3437   3465   3496   3597   3634   3669
O$APTS= 000000              8#     63
O$AU  = 000000              8#     63
O$BGNR= 000000              8#     63
O$BGNS= 000001              8#     56#    63
O$DU  = 000001              8#     56#    63
O$ERRT= 000000              8#     63
O$GNSW= 000001              8#     56#    63
O$POIN= 000001              8#     56#    63
O$SETU= 000000              8#     63    3764
PART1 = ****** U           59    691    741    789    827   1203   1311   1763   3718   3745
PART2 = 000001 G            2#     62    473    707   1206   1820   1971   2069   2238   2332   3714   3724   3736
                         3741   3750   3758
PASCNT  003236            508#   2689*  2757*  2758   2780   2796   2815   2834   3093*  3147   3149   3152*  3155*
                         3205*  3207   3209   3212   3214   3227   3231*  3359*  3428   3430   3433*  3436*  3481*
                         3483   3485   3487   3489   3492   3494   3509   3513*  3530*  3588   3590   3593*  3596*
                         3618   3620   3650*  3652   3654   3656   3658   3661   3676   3680*
PASNEW  014212           1262   1266#
PASNUM  003444            512#   1241*  1266*  3095   3112   3242   3361   3383   3532.  3550
PATTBL  002364            278#   2342
PAT1    005072            278     538#
PAT10   005346            287     632#
PAT2    005074            279     539#
PAT3    005134            280     556#
PAT4    005174            281     573#
PAT5    005234            282     590#
PAT6    005242            283     594#
PAT7    005302            284     611#
PAT8    005304            285     613#
PAT9    005344            286     630#
PH65$   020406           1833    1883#
PNT   = 001000 G           92#
POSHDS  020100           1823#
POSHD0  022176           2146    2148#
POSHSB  022172           2147#
POSHW1  022164           2145#
PRI   = 002000 G           92#
PRIOR = 000004             96#    3698
PRI00 = 000000 G           92#
PRI01 = 000040 G           92#
PRI02 = 000100 G           92#
PRI03 = 000140 G           92#
PRI04 = 000200 G           92#
PRI05   000240 G           92#
```

```
PRI06 = 000300 G          92#
PRI07 = 000340 G          92#
PSETNM  003446           513#  1248*  1269*  1270*  1273   1339   1370   1382   1499
PWCON   014462          1233   1259   1295   1305#
PWRFLG  003454           517#  1232*  1277   1279*  1323   1403   1405*
P2TO3E  006477           708#
P2T04E  006515           709#
P2T05E  006535           710#
P2T06E  006555           711#
P2T07E  006575           712#
P2T08E  006613           713#
P2T09E  006633           714#
P2T10E  006636           715#
P2T11E  006651           716#
P2T12E  006664           717#  2674
P2T13E  006676           718#  2864
P2T14E  006712           719#  2958
P2T15E  006733           720#  3006
P2T16E  006756           721#  3088
P2T17E  006777           722#  3249   3283
P2T18E  007031           723#  3354
P2T19E  007053           724#  3525
RDALHD  022756          2268#
RDDATA= 000114           125#  2435
RDHEAD= 000110           123#  2012   2705
RDNOHR= 000116           126#
RDYCHK  020444          1731   1897#  2006   2295   2445
RDYWAI  022222          1854   2159#  2694   2767   2786   2802   2824   2840   2878   2970   3026   3166   3172
                        3259   3446   3452   3606   3612
READRL  016172          1504#  1515   1526   1531   2303
RELDWT= 040000           145#  1575   1609
RESE3   010725           818#  1082   1155   1157   2634   3081
RESE4   010731           819#  1082   1155   1157   2635   3081
RESE5   010736           822#  2638
RESE6   010743           823#  1091   2400
RESPAR  003066           449#  2623   2655   2660
RESTAR  014154          1235   1254#
RESTBL  002324           260#  1062
REVSKO= 001000           140#   147   3663
REVSKS= 000200           138#   147   3500
RLBA  = 000002           157#  1505   1737*  2016*  2292*  2481*  3036*  3055*  3298*
RLBAS   003032           431#  1154   1274   1310   1337   1361   1366   1379   1481   1497   1978   2281   2646
                        2851   3080   3273   3338
RLCS  = 000000           156#  1013*  1015   1328*  1330   1362   1374*  1375   1396   1398*  1399   1524   1561
                        1612   1738*  2293   2297*  2299   2482*  2711*  2727*  3037*  3056*  3299*
RLCSR = 000000           162#  1504   1513   1601*  2017*
RLDA  = 000004           158#  1010*  1506   1600*  1736*  2015*  2291*  2480*  2710*  3035*  3054*  3297*
RLDRV   003036           433#  1012   1154   1326   1337   1366   1372   1379   1398   1497   1597   1716   1978
                        2010   2284   2449   2646   2706   2851   3080   3273   3283   3289   3338
RLMP  = 000006           159#  1019   1507   2043   2044   2282   2479*  3034*  3053*  3296*
RLVEC   003034           432#  1308   1402
RORWOP= 020000           144#  2574   2582   2585   2611
RPTOP   024662           927    939    953    967    982   1002   1102   1117   1138   1153   2564#
RPTREM  025656           931    945    959    974    989   1065   1108   1129   1144   2646#
RPTRES  025450           930    944    958    973    988   1063   1107   1128   1145   2620#
RSTRT   014072          1241#  1256
```

```
SAMSK = 000077            178#
SBSFIL  003502            532#  1980*  2530   2924   2937
SECQ    036676           3716   3743#
SECWD   007451            766#  2613
SEEK  = 000106            122#  1715
SEEKOP= 010000            143#  2574   2579   2608
SEQMES  007504            770#  2568
SETDON  014240           1253   1265   1270#
SFTPRM  036524 G         3711#
SGNWD   007440            764#  2610
SKTMES  007063            725#  2850
SPDSTA= 004000            207#  1571
SPTCOD  013720 G         1184#
SRTMES  007075            726#  3079
SSINDX  003006            418#  1229*  1551   1555*  1648*  1666   1670*  1751*  1824   1828*  1883*  1898   1902*
                         1931*  1990   1994*  2057*  2073   2077*  2130*  2160   2164*  2195*  2211   2215*  2227*
                         2242   2246*  2256*  2269   2273*  2319*  2377   2381*  2401*  2437   2441*  2509*  2565
                         2571
STAMES  007527            771#  1068
STAMSK= 000007            199#  1606
STATE2  010750            824#
STATE3  010760            825#
STATE5  010770            826#
STOSTA= 010000            208#
SUBSTK  002410            291#  1553*  1554*  1668*  1669*  1826*  1827*  1900*  1901*  1992*  1993*  2075*  2076*
                         2162*  2163*  2213*  2214*  2244*  2245*  2271*  2272*  2379*  2380*  2439*  2440*  2569
SVCBGL= 000001             11#
SVCGBL  000000             8#    58     63     66     67     90     224    645    877    922    936    950    964
                          979    994   1099   1113   1135   1148   1165   1173   1174   1184   1185   1202   1207
                         1215   1216   1357   1390   1391   1411   1420   2669   3691   3692   3711   3712   3764#
SVCINS= 000000             8#    12#    63     66     67    934    948    962    977    992   1035   1068   1082
                         1091   1097   1111   1133   1147   1154   1155   1157   1161   1174   1185   1207   1220
                         1221   1223   1224   1225   1226   1230   1231   1234   1235   1255   1256   1258   1259
                         1261   1262   1275   1276   1308   1309   1336   1337   1338   1339   1340   1343   1359
                         1365   1366   1368   1370   1377   1379   1381   1382   1383   1384   1393   1395   1402
                         1406   1407   1409   1413   1427   1428   1429   1434   1447   1448   1450   1454   1468
                         1484   1494   1495   1496   1497   1498   1499   1500   1583   1620   1633   1639   1744
                         1749   1835   1836   1837   1858   1874   1882   1917   1928   1977   1978   1979   2025
                         2036   2040   2048   2097   2109   2116   2178   2188   2192   2253   2306   2311   2400
                         2415   2473   2488   2494   2503   2507   2568   2569   2573   2586   2602   2606   2610
                         2613   2625   2634   2635   2638   2646   2648   2649   2650   2678   2718   2722   2850
                         2851   2852   2853   2854   2855   2856   2857   2858   2860   2873   2930   2931   2936
                         2943   2946   2954   2976   2986   2987   3001   3010   3012   3043   3047   3063   3067
                         3079   3080   3081   3083   3145   3224   3225   3238   3248   3273   3280   3283   3284
                         3307   3308   3314   3318   3325   3336   3338   3349   3426   3506   3507   3521   3586
                         3673   3674   3687   3692   3693   3694   3695   3696   3697   3698   3700   3712   3715
                         3716   3722   3725   3726   3727   3728   3729   3730   3731   3732   3733   3735   3737
                         3739   3764
SVCSUB= 000001             8#    10#  2976   3145   3248   3426   3586
SVCTAG= 000000             8#    13#   934    948    962    977    992   1097   1111   1133   1147   1161   1181
                         1199   1343   1384   1409   1413   1468   1484   1882   2860   2954   2986   3001   3083
                         3224   3238   3336   3349   3506   3521   3673   3687   3700   3739
SVCTST= 000001             8#     9#  2673   2863   2957   3005   3087   3241   3353   3524
SWAPHD  020744          1953#  2992   3232   3514   3681
S$LSYM= 010000             8#   934#   948#   962#   977#   992#  1097#  1111#  1133#  1147#  1161#  1181#  1199#
                         1343#  1384#  1409#  1413#  1468#  1484#  1835#  2860#  2954#  2986#  3001#  3083#  3224#
```

```
                    3238#  3336#  3349#  3506#  3521#  3673#  3687#  3700#  3739#
TAG      003470      523#
TBLS.R   003030      429#  3141*  3158   3229*  3234*  3423*  3511*  3516*  3583*  3678*  3683*
TBT      002550      331#  3134   3160   3411   3440   3576   3600
TCERR    007614      773#  1091   2400
TEMP     003464      521#  1457*  1482*
TEMP0    003122      464#  1437*  1482*  2086
TEMP1    003124      465#  1662*  1664*  1734   2425*  2427*  2466   2477
TEMP2    003126      466#
TEMP3    003130      467#  1005*  1020*  1021   1066   1068   2887*  2904*  2906   2920
TEMP4    003132      468#  1542   1543*  1545   1546*  1548   1549*  1559   1585   1594   1641   1652*  1986*
                    1988*  1999   2000
TEMP5    003134      469#  2883*  2888   2924*  2929*  2937
TEMP6    003136      470#  2884*  2889   2922   2925*  2932
TEMP7    003140      471#  2428*  2435*  2448
TEMP8    003142      472#  2705*  2706*  2707*  2727   2732
TIME     015424     1014   1423#  1602   1622   1739   1912   2018   2175   2483   2713   2729   3038   3057
                    3300
TIM.US   003466      522#  1424*  1442*
TOSLOW=  000001      151#
TRPFLG   003452      516#  1358*  1363   1466*
TRPHAN   015744     1359   1393   1466#
TSTINT   016362     1536#  2681   2865   2960   3015   3090   3250   3356   3527
TSTLAB   006471      690#  2573
TYPDR =  000006       97#  3697
T$ARGC=  000007       63#  1035#  1068#  1082#  1091#  1154#  1155#  1157#  1336#  1337#  1338#  1365#  1366#
                    1368#  1377#  1379#  1381#  1496#  1497#  1498#  1977#  1978#  1979#  2400#  2568#  2569#
                    2573#  2586#  2602#  2606#  2610#  2613#  2625#  2634#  2635#  2638#  2646#  2648#  2649#
                    2650#  2678#  2850#  2851#  2852#  2853#  2854#  2855#  2856#  2857#  2858#  3010#  3079#
                    3080#  3081#  3273#  3280#  3283#  3338#
T$CODE=  005052     3693#  3694#  3695#  3696#  3697#  3698#  3715#  3716#  3722#  3725#  3726#  3727#  3728#
                    3729#  3730#  3731#  3732#  3733#  3735#  3737#
T$ERRN=  003247       8#  1583#  1620#  1633#  1639#  1744#  1749#  1858#  1874#  1917#  1928#  2025#  2036#
                    2040#  2048#  2097#  2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#  2415#  2473#
                    2488#  2494#  2503#  2507#  2718#  2722#  2936#  2943#  2946#  3043#  3047#  3063#  3067#
                    3307#  3314#  3318#  3325#
T$EXCP=  000000     3694#  3695#  3696#  3698#  3727#  3730#  3733#  3735#  3737#
T$FLAG=  000040     2873#  2987#  3012#  3225#  3284#  3308#  3507#  3674#
T$GMAN=  000000       8#
T$HILI=  000377     3694#  3695#  3696#  3698#  3727#  3730#  3733#  3735#  3737#
T$LAST=  000001       8#  3764#
T$LOLI=  000001     3694#  3695#  3696#  3698#  3727#  3730#  3733#  3735#  3737#
T$LSYM=  010000       8#   934    948    962    977    992   1097   1111   1133   1147   1161   1181   1199
                    1343   1384   1409   1413   1468   1484   2860   2954   2986   3001   3083   3224   3238
                    3336   3349   3506   3521   3673   3687   3700   3739
T$LTNO=  000010     3764#
T$NEST=  177777       8#   58#   65#   90#  218#  224#  634#  645#  868#  877#  922#  934#  936#
                     948#  950#  962#  964#  977#  979#  992#  994#  1097#  1099#  1111#  1113#  1133#
                    1135#  1147#  1148#  1161#  1162#  1165#  1169#  1173#  1174#  1181#  1182#  1184#  1185#
                    1199#  1200#  1202#  1209#  1215#  1216#  1343#  1344#  1357#  1384#  1390#  1391#  1409#
                    1411#  1413#  1415#  1420#  1462#  1468#  1470#  1484#  1835#  1882#  2664#  2669#  2673#
                    2860#  2863#  2954#  2957#  2976#  2986#  3001#  3005#  3083#  3087#  3145#  3224#  3238#
                    3241#  3248#  3336#  3349#  3353#  3426#  3506#  3521#  3524#  3586#  3673#  3687#  3688#
                    3691#  3692#  3700#  3708#  3711#  3712#  3726   3729   3732   3739#  3762#
T$NSO    000000       58#   65    90#   218    224#   634    645#   868    877#   1162   1165#   1169   1173#
                    1182   1184#  1200   1202#  1209   1215#  1344   1357#  1384   1390#  1415   1420#  2664
```

```
                    2669#  3688   3691#  3708   3711#  3762
T$NS1 = 000005       922#   934    936#   948    950#   962   964#  977   979#  992   994#  1097  1099#
                    1111   1113#  1133   1135#  1147   1148#  1161  1174#  1181  1185#  1199  1216#  1343
                    1391#  1409   1411#  1413   1462#  1468   1470#  1484  1835#  1882   2673#  2860  2863#
                    2954   2957#  3001   3005#  3083   3087#  3238  3241#  3349  3353#  3521   3524#  3687
                    3692#  3700   3712#  3726   3729   3732   3739
T$NS2 = 000002      2976#  2986   3145#  3224   3248#  3336   3426#  3506  3586#  3673
T$PTNU= 000000         8#
T$SAVL= 177777         8#
T$SEGL= 177777         8#  1835#  1882#
T$SEK0= 010000      1835#  1882
T$SUBN= 000001         8#  2673#  2863#  2957#  2976#  3005#  3087#  3145#  3241#  3248#  3353#  3426#  3524#
                    3586#
T$TAGL= 177777         8#
T$TAGN= 010042         8#   922#   936#   950#   964#   979#   994#  1099#  1113#  1135#  1148#  1165#  1174#
                    1185#  1216#  1357#  1391#  1411#  1462#  1470#  2673#  2863#  2957#  2976#  3005#  3087#
                    3145#  3241#  3248#  3353#  3426#  3524#  3586#  3692#  3712#
T$TEMP= 000000        65#   218#   634#   868#   934#   948#   962#   977#   992#  1097#  1111#  1133#  1147#
                    1161#  1162#  1169#  1181#  1182#  1199#  1200#  1207#  1209#  1343#  1344#  1384#  1409#
                    1413#  1415#  1468#  1484#  1882#  2664#  2860#  2873#  2954#  2986#  2987#  3001#  3012#
                    3083#  3224#  3225#  3238#  3284#  3308#  3336#  3349#  3506#  3507#  3521#  3673#  3674#
                    3687#  3688#  3693#  3694#  3695#  3696#  3697#  3698#  3700#  3708#  3715#  3716#  3722#
                    3725#  3727#  3728#  3730#  3731#  3733#  3735#  3737#  3739#  3762#
T$TEST= 000010         8#  2673#  2863#  2957#  2976   3005#  3087#  3145   3241#  3248   3353#  3426   3524#
                    3586   3764
T$TSTM= 177777         8#   934    948    962    977    992   1035   1068   1082   1091   1097   1111   1133
                    1147   1154   1155   1157   1161   1220   1223   1224   1225   1230   1234   1255   1258
                    1261   1275   1308   1309   1336   1337   1338   1339   1340   1343   1359   1365   1366
                    1368   1370   1377   1379   1381   1382   1383   1384   1393   1395   1402   1406   1407
                    1409   1413   1427   1447   1494   1496   1497   1498   1499   1500   1583   1620   1633
                    1639   1744   1749   1835   1836   1858   1874   1882   1917   1928   1977   1978   1979
                    2025   2036   2040   2048   2097   2109   2116   2178   2188   2192   2253   2306   2311
                    2400   2415   2473   2488   2494   2503   2507   2568   2569   2573   2586   2602   2606
                    2610   2613   2625   2634   2635   2638   2646   2648   2649   2650   2678   2718   2722
                    2850   2851   2852   2853   2854   2855   2856   2857   2858   2860   2873   2930   2936
                    2943   2946   2954   2976   2986   2987   3001   3010   3012   3043   3047   3063   3067
                    3079   3080   3081   3083   3145   3224   3225   3238   3248   3273   3280   3283   3284
                    3307   3308   3314   3318   3325   3336   3338   3349   3426   3506   3507   3521   3586
                    3673   3674   3687
T$TSTS= 000001         8#  2673#  2863#  2957#  3005#  3087#  3241#  3353#  3524#
T$$AUT= 010016      1357#  1384
T$$CLE= 010017      1391#  1409
T$$DU = 010020      1411#  1413
T$$HAR= 010040      3692#  3700
T$$HW = 010013      1174#  1181
T$$INI= 010015      1216#  1343
T$$MSG= 010011       922#   934    936#   948    950#   962   964#  977   979#  992   994#  1097  1099#
                    1111   1113#  1133   1135#  1147   1148#  1161
T$$PRO= 010012      1165#
T$$SEG= 010000      1835#  1882#
T$$SOF= 010041      3712#  3739
T$$SRV= 010022      1462#  1468   1470#  1484
T$$SUB= 010037      2976#  2986   3145#  3224   3248#  3308   3336  3426#  3506  3586#  3673
T$$SW = 010014      1185#  1199
T$$TES= 010036      2673#  2860   2863#  2873   2954   2957#  2987  3001   3005#  3012   3083  3087#  3225
                    3238   3241#  3284   3349   3353#  3507   3521  3524#  3674   3687
```

```
T.BA     003052          440# 1477* 1505* 2650
T.CS     003050          439# 1006  1025  1069  1071  1476* 1504* 1519  1567  1577  1625  1631  1747
                        1856 1910  1921  1926  2021  2029  2038  2171  2182  2190  2309  2490  2498
                        2505 2650  2720  3045  3065  3311  3319* 3323
T.DA     003054          441# 1478* 1506* 2650
T.DRIV   002302          249# 1284* 1286  1685  1710  2761  2776  2811  2818  3115  3388  3555
T.MP     003056          442# 1023  1479* 1507* 1569  1571  1605  1627  2147  2650  3263  3278  3315
                        3320* 3321  3343
T.STAT   003064          447# 1105  1605* 1606*
T1       026142 G        1207 2673#
T2       030054 G        1207 2863#
T25TBL   002434          303#
T25TB2   002462          316#
T3       030572 G        1207 2957#
T3.1     030670          2976#
T306$    030636          2967# 2997
T3065$   031004          2962 2968  2971  2973  2993  3000#
T307$    030670          2975# 2991
T310$    030676          2966* 2974* 2978# 2988  2990* 2994*
T3100$   031740          3109# 3131  3133  3233
T3101$   032146          3144# 3236
T3165$   032722          3092  3126  3237# 3400
T3204$   033616          3337# 3342  3347
T3265$   033740          3246  3344  3348#
T33TBL   002510          330# 3101  3106* 3107* 3108* 3159  3229  3234  3368  3375* 3376* 3377* 3378*
                        3439 3511  3516  3539  3544* 3545* 3546* 3599  3678  3683
T3300$   034110          3379# 3408  3410  3515
T3301$   034372          3425# 3519
T3365$   035154          3358  3520#
T3400$   035310          3547# 3573  3575  3682
T3401$   035534          3585# 3685
T3465$   036346          3529  3565  3686#
T4       031006 G        1207 3005#
T5       031614 G        1207 3087#
T5.1     032146          3145#
T6       032724 G        1207 3241#
T6.1     032744          3248#
T7       033742 G        1207 3353#
T7.1     034372          3426#
T8       035156 G        1207 3524#
T8.1     035534          3586#
UAM   =  000200 G          92#
ULOAD =  000010           134#
UNDTST   007402           761#
UNXERR   006454           689# 1635
VALDES   007117           727# 2850  3079
VCNRST   006433           688# 1629
VCSTAT=  001000           205# 1627
VECMSG   036446          3695  3703#
VECT  =  000002            95# 3695
VERHDR   021626          2072#
VERPOS   022634          2241# 2696  2733  2769  2788  2804  2826  2842  2880  2972  3028  3174  3261
                        3454  3614
WAITIN   016224          1511# 1637  1742  2046  2304  2486  2716  3041  3061  3305
WCMSK =  017777           191#
WCRNG =  160000           192#
```

```
WDESTA= 100000  211#  1571
WGESTA= 002000  206#  3315  3320
WLSTAT= 020000  209#  3263  3278  3343
WRTSWI  003026  428#  3157* 3182  3190* 3194  3204* 3211* 3226* 3438* 3458  3466* 3470  3480*
                3491* 3503* 3508* 3598* 3627  3635* 3639  3649* 3660* 3670* 3675*
WTDATA= 000112  124#  2428  3288
XDELAY  003456  518#  1014* 1425  1426* 1430* 1435* 1482* 1602* 1622* 1739* 1912* 2018* 2175*
                2483* 2713* 2729* 3038* 3057* 3300*
XRDHD   021172  1988#  2218
XRDHDC  021162  1986#
XRDHDG  021176  1987  1989#
XREAD   024014  2435#  2892  2981  3196  3269  3331  3472  3641
XREADG  024022  2434  2436#
XSEEK   017326  1664#  1851  2691  2764  2783  2799  2821  2837  2875  2967  3023  3163  3169
                3256  3443  3449  3603  3609
XSEEKT  017316  1662#  2708
XSEEK1  017332  1663  1665#
XTIME   015570  1333  1401  1441#  1523  1563  1579  1614  1923  2184  3275  3340
XWRITE  023754  2427#  2979  3184  3267  3460  3629
XWRITT  023744  2425#  3032
XWRIT1  023760  2426  2428#
X$ALWA= 000000  8#
X$FALS= 000040  8#  3726  3729  3732
X$OFFS= 000400  8#  3726  3729  3732
X$TRUE= 000020  8#
YDELAY  003460  519#  1333* 1401* 1443  1444* 1445* 1446* 1451* 1455* 1482* 1523* 1563* 1579*
                1614* 1923* 2184* 3275* 3340*
.       - 037132  5#  66#  330#  331#  511#  532#  533#  535#  536#  871#  1429  1434  1450
                1454  2873  2987  3012  3225  3284  3308  3507  3674  3709#  3726  3729  3732
                3761#
```

```
ABORTW    30#  1482
BCOMPL   1226  1256  1259  1262  1276  1428  1495  2931
BGNAUT   1357
BGNCLN   1391
BGNDU    1411
BGNHRD   3692
BGNHW    1174
BGNINI   1216
BGNMOD     58    90   224   645   877  1173  1184  1202  1215  1390  1420  2669  3691  3711
BGNMSG    922   936   950   964   979   994  1099  1113  1135  1148
BGNPRO   1165
BGNSEG   1835
BGNSFT   3712
BGNSRV   1462  1470
BGNSUB   2976  3145  3248  3426  3586
BGNSW    1185
BGNTST   2673  2863  2957  3005  3087  3241  3353  3524
BNCOMP   1221  1231  1235  1448  1937
BRESET   1224  1407
CLOCK    1220
CLRVEC   1383  1402  1406
DELAY    1429  1434  1450  1454
DESCRI     66
DEVTYP     67
DISPAT   1207
DOCLN    1340  1500
DODU     1339  1370  1382  1499
ENDAUT   1384
ENDCLN   1409
ENDDU    1413
ENDHRD   3700
ENDHW    1181
ENDINI   1343
ENDMOD     65   218   634   868  1162  1182  1200  1209  1344  1415  2664  3688  3708  3762
ENDMSG    934   948   962   977   992  1097  1111  1133  1147  1161
ENDPRO   1169
ENDSEG   1882
ENDSFT   3739
ENDSRV   1468  1484
ENDSUB   2986  3224  3336  3506  3673
ENDSW    1199
ENDTST   2860  2954  3001  3083  3238  3349  3521  3687
EQUALS     92
ERRHRD   1583  1620  1633  1639  1744  1749  1858  1874  1917  1928  2025  2036  2040  2048  2097
         2109  2116  2178  2188  2192  2253  2306  2311  2415  2473  2488  2494  2503  2507  2718
         2722  2936  2943  2946  3043  3047  3063  3067  3307  3314  3318  3325
ESCAPE   2987  3225  3507  3674
EXIT     2873  3012  3284  3308
GETTIM    37#  2730  3058
GPHARD   1275
GPRMA    3694  3695
GPRMD    3696  3698  3727  3730  3733  3735  3737
GPRML    3693  3697  3715  3716  3722  3725  3728  3731
HEADER     63
INLOOP   1494  1836  2930
LASTAD   3764
```

```
MANUAL  1225
MSBYTE    63#
MSCHEC  2873#  3012#  3284#  3308#
MSCNTO  3693#  3694#  3695#  3696#  3697#  3698#  3715#  3716#  3722#  3725#  3727#  3728#  3730#  3731#  3733#
        3735#  3737#
MSCOUN  1035#  1068#  1082#  1091#  1154#  1155#  1157#  1336#  1337#  1338#  1365#  1366#  1368#  1377#  1379#
        1381#  1496#  1497#  1498#  1977#  1978#  1979#  2400#  2568#  2569#  2573#  2586#  2602#  2606#  2610#
        2613#  2625#  2634#  2635#  2638#  2646#  2648#  2649#  2650#  2678#  2850#  2851#  2852#  2853#  2854#
        2855#  2856#  2857#  2858#  3010#  3079#  3080#  3081#  3273#  3280#  3283#  3338#
MSDATA    63#    66#    67#
MSDECR    65#   218#   634#   868#   934#   948#   962#   977#   992#  1097#  1111#  1133#  1147#  1161#  1162#
        1169#  1181#  1182#  1199#  1200#  1209#  1343#  1344#  1384#  1409#  1413#  1415#  1468#  1484#  1882#
        2664#  2860#  2954#  2986#  3001#  3083#  3224#  3238#  3336#  3349#  3506#  3521#  3673#  3687#  3688#
        3700#  3708#  3739#  3762#
MSDEFA  3693#  3694#  3695#  3696#  3697#  3698#  3715#  3716#  3722#  3725#  3727#  3728#  3730#  3731#  3733#
        3735#  3737#
MSENDE    65#   218#   634#   868#   934#   948#   962#   977#   992#  1097#  1111#  1133#  1147#  1161#  1162#
        1181#  1182#  1199#  1200#  1209#  1343#  1344#  1384#  1409#  1413#  1415#  1468#  1484#  1882#  2664#
        2860#  2954#  2986#  3001#  3083#  3224#  3238#  3336#  3349#  3506#  3521#  3673#  3687#  3688#  3700#
        3708#  3739#  3762#
MSERRI  1583#  1620#  1633#  1639#  1744#  1749#  1858#  1874#  1917#  1928#  2025#  2036#  2040#  2048#  2097#
        2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#  2415#  2473#  2488#  2494#  2503#  2507#  2718#
        2722#  2936#  2943#  2946#  3043#  3047#  3063#  3067#  3307#  3314#  3318#  3325#
MSESCA  2987#  3225#  3507#  3674#
MSESCS  2987#  3225#  3507#  3674#
MSEXCP  3694#  3695#  3696#  3698#  3727#  3730#  3733#  3735#  3737#
MSEXIT  2873#  3012#  3284#  3308#
MSEXSE  2873#  3012#  3284#  3308#
MSEXTJ  2873#  3012#  3284#  3308#
MSGEN     58#    63#    66#    67#    90#   224#   645#   877#   922#   934#   936#   948#   950#   962#   964#
        977#   979#   992#   994#  1097#  1099#  1111#  1113#  1133#  1135#  1147#  1148#  1161#  1165#  1173#
        1174#  1181#  1184#  1185#  1199#  1202#  1207#  1215#  1216#  1343#  1357#  1384#  1390#  1391#  1409#
        1411#  1413#  1420#  1462#  1468#  1470#  1484#  1882#  2669#  2673#  2860#  2863#  2954#  2957#  2976#
        2986#  3001#  3005#  3083#  3087#  3145#  3224#  3238#  3241#  3248#  3336#  3349#  3353#  3426#  3506#
        3521#  3524#  3586#  3673#  3687#  3691#  3692#  3700#  3711#  3712#  3739#  3764#
MSGETS    65#   218#   634#   868#   934#   948#   962#   977#   992#  1097#  1111#  1133#  1147#  1161#  1162#
        1169#  1181#  1182#  1199#  1200#  1209#  1343#  1344#  1384#  1409#  1413#  1415#  1468#  1484#  1882#
        2664#  2860#  2954#  2986#  3001#  3083#  3224#  3238#  3336#  3349#  3506#  3521#  3673#  3687#  3688#
        3700#  3708#  3726#  3729#  3732#  3739#  3762#
MSGETT  2873#  2987#  3012#  3225#  3284#  3308#  3507#  3674#  3726#  3729#  3732#
MSGNGB    58#    63#    66#    67#    90#   224#   645#   877#   922#   936#   950#   964#   979#   994#  1099#
        1113#  1135#  1148#  1165#  1173#  1174#  1184#  1185#  1202#  1207#  1215#  1216#  1357#  1390#  1391#
        1411#  1420#  1462#  1470#  2669#  3691#  3692#  3711#  3712#  3764#
MSGNIN    63#    66#    67#   934#   948#   962#   977#   992#  1035#  1068#  1082#  1091#  1097#  1111#  1133#
        1147#  1154#  1155#  1157#  1161#  1174#  1185#  1207#  1220#  1221#  1223#  1224#  1225#  1226#  1230#
        1231#  1234#  1235#  1255#  1256#  1258#  1259#  1261#  1262#  1275#  1276#  1308#  1309#  1336#  1337#
        1338#  1339#  1340#  1343#  1359#  1365#  1366#  1368#  1370#  1377#  1379#  1381#  1382#  1383#  1384#
        1393#  1395#  1402#  1406#  1407#  1409#  1413#  1427#  1428#  1429#  1434#  1447#  1448#  1450#  1454#
        1468#  1484#  1494#  1495#  1496#  1497#  1498#  1499#  1500#  1583#  1620#  1633#  1639#  1744#  1749#
        1835#  1836#  1837#  1858#  1874#  1882#  1917#  1928#  1977#  1978#  1979#  2025#  2036#  2040#  2048#
        2097#  2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#  2400#  2415#  2473#  2488#  2494#  2503#
        2507#  2568#  2569#  2573#  2586#  2602#  2606#  2610#  2613#  2625#  2634#  2635#  2638#  2646#  2648#
        2649#  2650#  2678#  2718#  2722#  2850#  2851#  2852#  2853#  2854#  2855#  2856#  2857#  2858#  2860#
        2873#  2930#  2931#  2936#  2943#  2946#  2954#  2976#  2986#  2987#  3001#  3010#  3012#  3043#  3047#
        3063#  3067#  3079#  3080#  3081#  3083#  3145#  3224#  3225#  3238#  3248#  3273#  3280#  3283#  3284#
        3307#  3308#  3314#  3318#  3325#  3336#  3338#  3349#  3426#  3506#  3507#  3521#  3586#  3673#  3674#
```

	3687#	3692#	3693#	3694#	3695#	3696#	3697#	3698#	3700#	3712#	3715#	3716#	3722#	3725#	3726#
	3727#	3728#	3729#	3730#	3731#	3732#	3733#	3735#	3737#	3739#	3764#				
M$GNLS	1882#														
M$GNSU	2976#	3145#	3248#	3426#	3586#										
M$GNTA	934#	948#	962#	977#	992#	1097#	1111#	1133#	1147#	1161#	1181#	1199#	1343#	1384#	1409#
	1413#	1468#	1484#	2860#	2954#	2986#	3001#	3083#	3224#	3238#	3336#	3349#	3506#	3521#	3673#
	3687#	3700#	3739#												
M$GNTE	2673#	2863#	2957#	3005#	3087#	3241#	3353#	3524#							
M$HAPT	63#														
M$HNAP	63#														
M$INCR	58#	90#	224#	645#	877#	922#	934#	936#	948#	950#	962#	964#	977#	979#	992#
	994#	1035#	1068#	1082#	1091#	1097#	1099#	1111#	1113#	1133#	1135#	1147#	1148#	1154#	1155#
	1157#	1161#	1165#	1173#	1174#	1184#	1185#	1202#	1215#	1216#	1220#	1223#	1224#	1225#	1230#
	1234#	1255#	1258#	1261#	1275#	1308#	1309#	1336#	1337#	1338#	1339#	1340#	1343#	1357#	1359#
	1365#	1366#	1368#	1370#	1377#	1379#	1381#	1382#	1383#	1384#	1390#	1391#	1393#	1395#	1402#
	1406#	1407#	1409#	1411#	1413#	1420#	1427#	1447#	1462#	1470#	1494#	1496#	1497#	1498#	1499#
	1500#	1583#	1620#	1633#	1639#	1744#	1749#	1835#	1836#	1858#	1874#	1882#	1917#	1928#	1977#
	1978#	1979#	2025#	2036#	2040#	2048#	2097#	2109#	2116#	2178#	2188#	2192#	2253#	2306#	2311#
	2400#	2415#	2473#	2488#	2494#	2503#	2507#	2568#	2569#	2573#	2586#	2602#	2606#	2610#	2613#
	2625#	2634#	2635#	2638#	2646#	2648#	2649#	2650#	2669#	2673#	2678#	2718#	2722#	2850#	2851#
	2852#	2853#	2854#	2855#	2856#	2857#	2858#	2860#	2863#	2873#	2930#	2936#	2943#	2946#	2954#
	2957#	2976#	2986#	2987#	3001#	3005#	3010#	3012#	3043#	3047#	3063#	3067#	3079#	3080#	3081#
	3083#	3087#	3145#	3224#	3225#	3238#	3241#	3248#	3273#	3280#	3283#	3284#	3307#	3308#	3314#
	3318#	3325#	3336#	3338#	3349#	3353#	3426#	3506#	3507#	3521#	3524#	3586#	3673#	3674#	3687#
	3691#	3692#	3711#	3712#											
M$LDRO	1220#	1223#	1230#	1234#	1255#	1258#	1261#	1275#	1309#	1339#	1370#	1382#	1383#	1395#	1402#
	1406#	1499#													
M$MCHI	8#														
M$MCLO	8#														
M$POP	65#	218#	634#	868#	934#	948#	962#	977#	992#	1097#	1111#	1133#	1147#	1161#	1162#
	1169#	1181#	1182#	1199#	1200#	1209#	1343#	1344#	1384#	1409#	1413#	1415#	1468#	1484#	1882#
	2664#	2860#	2954#	2986#	3001#	3083#	3224#	3238#	3336#	3349#	3506#	3521#	3673#	3687#	3688#
	3700#	3708#	3739#	3762#											
M$PRIN	1035#	1068#	1082#	1091#	1154#	1155#	1157#	1336#	1337#	1338#	1365#	1366#	1368#	1377#	1379#
	1381#	1496#	1497#	1498#	1977#	1978#	1979#	2400#	2568#	2569#	2573#	2586#	2602#	2606#	2610#
	2613#	2625#	2634#	2635#	2638#	2646#	2648#	2649#	2650#	2678#	2850#	2851#	2852#	2853#	2854#
	2855#	2856#	2857#	2858#	3010#	3079#	3080#	3081#	3273#	3280#	3283#	3338#			
M$PUSH	58#	90#	224#	645#	877#	922#	936#	950#	964#	979#	994#	1099#	1113#	1135#	1148#
	1165#	1173#	1174#	1184#	1185#	1202#	1215#	1216#	1357#	1390#	1391#	1411#	1420#	1462#	1470#
	1835#	2669#	2673#	2863#	2957#	2976#	3005#	3087#	3145#	3241#	3248#	3353#	3426#	3524#	3586#
	3691#	3692#	3711#	3712#											
M$PUT	1035#	1068#	1082#	1091#	1154#	1155#	1157#	1308#	1336#	1337#	1338#	1359#	1365#	1366#	1368#
	1377#	1379#	1381#	1393#	1496#	1497#	1498#	1977#	1978#	1979#	2400#	2568#	2569#	2573#	2586#
	2602#	2606#	2610#	2613#	2625#	2634#	2635#	2638#	2646#	2648#	2649#	2650#	2678#	2850#	2851#
	2852#	2853#	2854#	2855#	2856#	2857#	2858#	3010#	3079#	3080#	3081#	3273#	3280#	3283#	3338#
M$PUT1	1035#	1068#	1082#	1091#	1154#	1155#	1157#	1308#	1336#	1337#	1338#	1359#	1365#	1366#	1368#
	1377#	1379#	1381#	1393#	1496#	1497#	1498#	1977#	1978#	1979#	2400#	2568#	2569#	2573#	2586#
	2602#	2606#	2610#	2613#	2625#	2634#	2635#	2638#	2646#	2648#	2649#	2650#	2678#	2850#	2851#
	2852#	2853#	2854#	2855#	2856#	2857#	2858#	3010#	3079#	3080#	3081#	3273#	3280#	3283#	3338#
M$RADI	3693#	3694#	3695#	3696#	3697#	3698#	3715#	3716#	3722#	3725#	3727#	3728#	3730#	3731#	3733#
	3735#	3737#													
M$RNRO	1220#	1275#													
M$SETS	58#	90#	224#	645#	877#	922#	936#	950#	964#	979#	994#	1099#	1113#	1135#	1148#
	1165#	1173#	1174#	1184#	1185#	1202#	1215#	1216#	1357#	1390#	1391#	1411#	1420#	1462#	1470#
	1835#	2669#	2673#	2863#	2957#	2976#	3005#	3087#	3145#	3241#	3248#	3353#	3426#	3524#	3586#
	3691#	3692#	3711#	3712#											

```
M$SVC     934#   948#   962#   977#   992#  1035#  1068#  1082#  1091#  1097#  1111#  1133#  1147#  1154#  1155#
         1157#  1161#  1220#  1223#  1224#  1225#  1230#  1234#  1255#  1258#  1261#  1275#  1308#  1309#  1336#
         1337#  1338#  1339#  1340#  1343#  1359#  1365#  1366#  1368#  1370#  1377#  1379#  1381#  1382#  1383#
         1384#  1393#  1395#  1402#  1406#  1407#  1409#  1413#  1427#  1447#  1494#  1496#  1497#  1498#  1499#
         1500#  1583   1620   1633   1639   1744   1749   1835#  1836#  1858   1874   1882#  1917   1928   1977#
         1978#  1979#  2025   2036   2040   2048   2097   2109   2116   2178   2188   2192   2253   2306   2311
         2400#  2415   2473   2488   2494   2503   2507   2568#  2569#  2573#  2586#  2602#  2606#  2610#  2613#
         2625#  2634#  2635#  2638#  2646#  2648#  2649#  2650#  2678#  2718   2722   2850#  2851#  2852#  2853#
         2854#  2855#  2856#  2857#  2858#  2860#  2873#  2930#  2936   2943   2946   2954#  2976#  2986#  2987#
         3001#  3010#  3012#  3043   3047   3063   3067   3079#  3080#  3081#  3083#  3145#  3224#  3225#  3238#
         3248#  3273#  3280#  3283#  3284#  3307   3308#  3314   3318   3325   3336#  3338#  3349#  3426#  3506#
         3507#  3521#  3586#  3673#  3674#  3687#
M$TLAB    934#   948#   962#   977#   992#  1035#  1068#  1082#  1091#  1097#  1111#  1133#  1147#  1154#  1155#
         1157#  1161#  1220#  1223#  1224#  1225#  1230#  1234#  1255#  1258#  1261#  1275#  1308#  1309#  1336#
         1337#  1338#  1339#  1340#  1343#  1359#  1365#  1366#  1368#  1370#  1377#  1379#  1381#  1382#  1383#
         1384#  1393#  1395#  1402#  1406#  1407#  1409#  1413#  1427#  1447#  1494#  1496#  1497#  1498#  1499#
         1500#  1583#  1620#  1633#  1639#  1744#  1749#  1835#  1836#  1858#  1874#  1882#  1917#  1928#  1977#
         1978#  1979#  2025#  2036#  2040#  2048#  2097#  2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#
         2400#  2415#  2473#  2488#  2494#  2503#  2507#  2568#  2569#  2573#  2586#  2602#  2606#  2610#  2613#
         2625#  2634#  2635#  2638#  2646#  2648#  2649#  2650#  2678#  2718#  2722#  2850#  2851#  2852#  2853#
         2854#  2855#  2856#  2857#  2858#  2860#  2873#  2930#  2936#  2943#  2946#  2954#  2976#  2986#  2987#
         3001#  3010#  3012#  3043#  3047#  3063#  3067#  3079#  3080#  3081#  3083#  3145#  3224#  3225#  3238#
         3248#  3273#  3280#  3283#  3284#  3307#  3308#  3314#  3318#  3325#  3336#  3338#  3349#  3426#  3506#
         3507#  3521#  3586#  3673#  3674#  3687#
M$TSTL    934#   948#   962#   977#   992#  1035#  1068#  1082#  1091#  1097#  1111#  1133#  1147#  1154#  1155#
         1157#  1161#  1220#  1223#  1224#  1225#  1230#  1234#  1255#  1258#  1261#  1275#  1308#  1309#  1336#
         1337#  1338#  1339#  1340#  1343#  1359#  1365#  1366#  1368#  1370#  1377#  1379#  1381#  1382#  1383#
         1384#  1393#  1395#  1402#  1406#  1407#  1409#  1413#  1427#  1447#  1494#  1496#  1497#  1498#  1499#
         1500#  1583#  1620#  1633#  1639#  1744#  1749#  1835#  1836#  1858#  1874#  1882#  1917#  1928#  1977#
         1978#  1979#  2025#  2036#  2040#  2048#  2097#  2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#
         2400#  2415#  2473#  2488#  2494#  2503#  2507#  2568#  2569#  2573#  2586#  2602#  2606#  2610#  2613#
         2625#  2634#  2635#  2638#  2646#  2648#  2649#  2650#  2678#  2718#  2722#  2850#  2851#  2852#  2853#
         2854#  2855#  2856#  2857#  2858#  2860#  2873#  2930#  2936#  2943#  2946#  2954#  2976#  2986#  2987#
         3001#  3010#  3012#  3043#  3047#  3063#  3067#  3079#  3080#  3081#  3083#  3145#  3224#  3225#  3238#
         3248#  3273#  3280#  3283#  3284#  3307#  3308#  3314#  3318#  3325#  3336#  3338#  3349#  3426#  3506#
         3507#  3521#  3586#  3673#  3674#  3687#
M$WORD      63#  1207#  1583#  1620#  1633#  1639#  1744#  1749#  1858#  1874#  1917#  1928#  2025#  2036#  2040#
         2048#  2097#  2109#  2116#  2178#  2188#  2192#  2253#  2306#  2311#  2415#  2473#  2488#  2494#  2503#
         2507#  2718#  2722#  2873#  2936#  2943#  2946#  3012#  3043#  3047#  3063#  3067#  3284#  3307#  3308#
         3314#  3318#  3325#  3693#  3694#  3695#  3696#  3697#  3698#  3715#  3716#  3722#  3725#  3726#  3727#
         3728#  3729#  3730#  3731#  3732#  3733#  3735#  3737#  3764
M$XFER    3726#  3729#  3732#
POINTE      56
PRINTB    1035   1068   1082   1091   1154   1155   1157   2400   2568   2569   2573   2586   2602   2606   2610
          2613   2625   2634   2635   2638   2646   2648   2649   2650
PRINTF    1336   1337   1338   1365   1366   1368   1377   1379   1381   1496   1497   1498   1977   1978   1979
          2678   2850   2851   2852   2853   2854   2855   2856   2857   2858   3010   3079   3080   3081   3273
          3280   3283   3338
READBU    1427   1447
READEF    1230   1234   1255   1258   1261
SETPRI    1223   1309   1395
SETVEC    1308   1359   1393
STCLK       42#  2725   3051
SVC          6#     8
WAITMS      25#  1333   1401   1523   1563   1579   1614   1923   2184   3275   3340
WAITUS      20#  1014   1602   1622   1739   1912   2018   2175   2483   2713   2729   3038   3057   3300
```

```
XFER     2873#   3012#   3284#   3308#
XFERF    3726    3729    3732


. ABS.  037132     000

 ERRORS DETECTED:  0

 ,CZRLNA.LST/CRF=SVC33/ML,CZRLNA.MAC
 RUN-TIME: 109 109 11 SECONDS
 RUN-TIME RATIO: 462/231-1.9
 CORE USED:  16K  (31 PAGES)
```